Broadview

——GNU 宣言

Linux管理与开发实用指南
——基于Ubuntu

叶乐 等编著

本书特点

- 循序渐进，由浅入深，掌握Linux的基本操作与技能；
- 内容丰富，适合各级Linux读者，初学者可了解实用的Linux软件知识，中级用户可从Shell命令参数表及使用方法中得到许多有用的资料，系统管理员可学到系统管理和性能优化方面的技巧。

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

B

Linux管理与开发实用指南——基于Ubuntu

马广飞　等编著



電子工業出版社
Publishing House of Electronics Industry
北京・BEIJING

内 容 简 介

本书内容由浅入深，介绍了 Linux 的入门、应用软件使用及系统管理各层次的内容，包括 Linux 的基本概念、Linux 的安装详解、Linux 桌面环境、重要的文本编辑器 vim、Emacs 的使用、正则表达式的基本知识、数百个常用 Shell 命令的使用，以及常用软件（如办公软件、多媒体软件等）的使用。对于程序开发人员，本书介绍了 Linux 环境中重要的编译器 GCC 的简单使用、GDB 调试器的使用和 Shell 编程的知识。对于系统管理员，本书介绍了系统安全、网络管理、硬件管理，以及用户与组管理等方面的内容，可以满足初步系统管理的需求。

本书适合广大 Linux 爱好者、Linux 环境的初级开发人员、Linux 初级系统管理员阅读，同时也可作为大专院校及培训机构的教材。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

Linux 管理与开发实用指南：基于 Ubuntu / 马广飞等编著. —北京：电子工业出版社，2009.10
ISBN 978-7-121-09650-1

I. L… Ⅱ. 马… Ⅲ. Linux 操作系统－指南 Ⅳ. TP316.89-62

中国版本图书馆 CIP 数据核字（2009）第 180384 号

责任编辑：顾慧芳
印　　刷：北京智力达印刷有限公司
装　　订：北京中新伟业印刷有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：787×1092　1/16　印张：34　字数：778 千字
印　　次：2009 年 10 月第 1 次印刷
印　　数：4000 册　定价：59.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

作为一个基于 GNU/Linux 的平台，Ubuntu*操作系统将 ubuntu 精神带到了软件世界。

——GNU 宣言

1991 年 10 月 5 日，Linus 在 comp.os.minix 新闻组上公布自己的 Linux 操作系统内核时，恐怕没有人会相信十几年后的今天，Linux 会将自己的触角延伸到大型机、各种服务器、个人电脑、手机，以及各种嵌入式系统。而且，经过全球开发者与大型开源公司的努力，Linux 已经逐步拥有了华丽的界面，友好的人机交互模式，在很大程度上降低了使用 Linux 系统的门槛。

2008 年番茄花园网站的主人被捕，以及年末微软的“黑屏”事件，使得普通用户对于版权这个原来不怎么在意的东西慢慢多了一份关注的目光。他们发现，原来操作系统世界中，除了 Windows，还有一个同样优秀而且免费的成员——Linux。稍微对 Linux 多一些了解后，许多人会为其强大的性能所折服，原来在 Windows 世界之外，还有一个同样精彩的 Linux 世界。这个世界没有游戏的嘈杂，也没有版权的纠葛，只有自由、开源的精神，更有着强大的软件，从普通用户可能用到的办公软件、影音播放处理软件，到专业的科学计算、绘图、排版软件，以及编程、网页制作软件等，可以满足日常工作、生活的绝大部分需要。

但由于种种原因，Linux 在个人桌面市场只占据了很少的市场份额，这个局面的造成有着很复杂的原因，其中一个原因就是 Linux 的市场推广不够。为了让 Linux 被更多的人所了解和掌握，编者花费了大量的精力和时间编写了这本介绍 Linux 系统的书籍。通过对本书的学习，用户应该可以掌握 Linux 的基本操作与技能，并且初步具备系统管理的知识。

本书的特点

1．循序渐进，由浅入深

本书先介绍了 Linux 的基本概念，让读者对 Linux 系统有了初步的认识，接着介绍了桌面系统的

* Ubuntu 是一个南非的民族观念，着眼于人们之间的忠诚和联系。该词来自于祖鲁语和科萨语。Ubuntu（发音“oo-BOON-too”—“乌班图”，内置视频中曼德拉的发音为/u:bu:tu:/全部 u 发音/wu:/）被视为非洲人的传统理念，也是建立新南非共和国的基本原则之一，与非洲复兴的理想密切相关。ubuntu 精神的大意是“人道待人”（对他人仁慈）。另一种翻译可以是：“天下共享的信念，连接起每个人”。“具有 ubuntu 精神的人心胸开阔，乐于助人，见贤思齐而不忌妒贤能，因为他/她拥有适度的自信，而这源自如下认识：自己乃是属于一个更大的整体，当他人受到伤害或死去时，当他人受到折磨或压迫时，这个整体就会消失。”——大主教 Desmond Tutu。作为一个基于 GNU/Linux 的平台，Ubuntu 操作系统将 ubuntu 精神带到了软件世界。

使用，然后依次介绍了 Shell 命令等高阶知识。这样，对于初学者来说是一个平滑的学习过程，可以顺利地掌握 Linux 的各层次知识。

2．内容丰富，适合各级读者

本书除了简单介绍 Linux 概念外，将大量的篇幅用于介绍 Linux 实用的软件操作与技能。对于初学者而言，可以从桌面系统和应用软件开始阅读。而对于中级用户而言，可以从丰富的 Shell 命令参数表及使用技巧中得到许多有用的资料，也可以学习到从正则表达式到 Shell 编程等各方面的技能。系统管理员则可以从系统管理篇中得到系统管理，性能优化等各方面的知识。

3．由表及里，深入介绍相关知识

授人以鱼，不如授人以渔。所以本书在介绍相关知识点时，不满足于让读者知道该怎么做，还介绍了丰富的背景知识，让读者理解为什么要这么做。这样，在遇到相关问题时，读者就可以很轻易地运用相关背景知识分析得到最佳的解决方案，灵活应对各种复杂　　问题。

4．提供良好的技术支持

如果在本书的阅读过程中发现任何问题，或者在实践中出现技术性的障碍，可以登录本书专属论坛，发表帖子，我或者其他读者将会尽可能解决您的问题。

本书的内容

本书可分为五篇共 22 章，第一篇 Linux 基础篇包括第 1～5 章，第二篇 Linux 系统管理篇包括第 6～10 章，第三篇 Linux 应用篇包括第 11～16 章，第四篇 Linux 开发篇包括第 17～20 章，第五篇 Linux 安全篇包括第 21～22 章。具体内容如下。

第 1 章主要介绍了 Linux 系统的基本概念，向读者介绍 Linux 的特点与独特的文化，使读者对 Linux 有一个概念上的认识。

第 2 章介绍了 Ubuntu 的安装方法与详细的安装过程，使得没有 Linux 安装经验的读者可以轻松地进行安装，开始 Linux 的学习、使用过程。

第 3 章介绍了对于普通用户来说最为重要的桌面环境的使用方法。通过桌面系统的学习，读者应该可以完成大多数日常工作。

第 4 章是一些在实际使用前必须完成的配置项，如宽带拨号的设置、中文输入法的安装等，完成这些设置就可以正常使用 Linux 系统了。

第 5 章介绍了对于 Linux 来说，必须要掌握的文本编辑器 vim 与 Emacs 的使用方法，以及正则表达式的简介。正则表达式与文本编辑器则为编辑配置文件打下了良好的基础。

第 6 章介绍了 Shell 的基本命令。Shell 命令对于 Linux 用户来说是最为重要的内容之一，掌握 Linux 的使用基本上也就是掌握这些命令的用法。本章不但介绍了大多数常用命令，还列出了命令的许多参数，可供使用时查阅。

第 7 章介绍了 Linux 的文件系统与目录结构，方便读者理解 Linux 与 Windows 的不同点。本章还介绍了挂载文件系统的方法，面向入门用户介绍了文件管理器的使用，以及压缩文件管理等内容。

第 8 章的主要内容是软件包的管理，也就是说，如何在 Linux 中安装与卸载软件的方法，通过本章的学习，读者应具备在 Linux 系统中进行软件管理的技能。

第 9 章介绍了对磁盘进行分区与格式化等管理的方法，包括在未安装系统前如何进行硬盘分区等操作，以及查看硬盘使用情况的方法。还介绍了使用 dd 命令进行硬盘复制等内容。

第 10 章介绍了进程管理的内容，包括显示进程、中止进程的内容，还介绍了进程管理的高级内容，如进程的后台执行、进程的自动调度等方面的内容。

第 11 章介绍了几个网络管理命令的使用，以及对于 Linux 网络管理而言极其重要的基础知识 iptables 的概念及使用，还介绍了几种远程登录的方法，以便于读者进行各种远程登录的操作。

第 12 章介绍了 Linux 世界中最常见的网络浏览器 FireFox 及 Opera 的使用及一些设置技巧，同时介绍了如何在 Linux 中安装 IE 的方法。本章还介绍了 Linux 世界中很有特色的字符界面网络浏览软件与电子邮件软件。本章的内容还包括一些强大的即时通信软件的相关内容。

第 13 章介绍了影音播放软件的简单使用，包括著名的播放器 Myplayer 等视频播放软件，以及 Audacious 等音频播放软件，还介绍了一些视频编辑软件的使用，如 LIVES、Avidemux 等。

第 14 章介绍了图形浏览软件、图片管理软件的使用，以及著名的图像处理软件 GIMP 和强大的矢量图绘制软件的使用方法。

第 15 章介绍了办公软件的使用，主要介绍了 OpenOffice.org 的使用，包括文字处理、电子表格、幻灯片制作等功能；同时简单介绍了其他一些办公软件的使用，以及中文 PDF、CHM 文件的打开方法。

第 16 章介绍了专业排版软件 LaTeX 的知识，由于 LaTeX 是世界科技论文的标准处理格式，同时其功能也比较强大，了解这方面的知识有利于与全球科研人士进行交流，也可以排出更为专业美观的版面。

第 17 章介绍了 Linux 世界中重要的编译器 **GCC** 的使用，以及使用 **GDB** 调试的方法，makefile 文件的写法等，还简单介绍了一款比较流行的开发环境 Eclipse。

第 18 章介绍了 Shell 编程的基本知识。本章内容是从零开始介绍了 Shell 编辑的知识。因为 Shell 功能强大而又使用简单，对于普通用户而言，可能并无其他的编程经验，故使用 Shell 编程的几率更大些。

第 19 章介绍了 LAMP 服务器的搭建过程及基本设置，以及通过两个 Web 程序（个人博客系统、维基百科）的安装说明了该服务器的功能。

第 20 章以 WU-FTP 服务器软件为例介绍了 FTP 服务器的搭建及设置，以及 FTP 登录软件的使用，一个是图形界面的 FileZilla，另一个是字符界面的 lftp。

第 21 章介绍了用户与组管理的知识，包括用户与组创建、删除与管理，同时以用户与组的配置文件为基础介绍了配置账户、密码及权限的方法。

第 22 章介绍了基本的安全常识，包括一个使用简单而又功能强大的防火墙软件 FireStarter 的使用，以及 Linux 系统中基本加密系统 PAM 的原理，包括使用顶级加密技术进行数据加密的方法；另外，还介绍了通过配置文件进行系统安全分析的基本知识。

适合的读者

- Linux 爱好者
- Linux 程序开发人员

- Linux 系统管理员
- 大中专院校的学生
- 培训机构

本书作者

本书由马广飞主笔编写，其他参与编写的人员还有昊燃、黄叶新、李增辉、刘聪、龙志标、孟劼、米爱中、秦侠、冉剑、任潇、谭励、王大亮、伍云辉、于重重、曾光、张双、朱照华、扈新波、黄永湛、刘新、孟祥嵩、王二亮、张贺军、李勇、刘基林、闫术卓、关涛、王岩、李晓白、魏星、刘蕾、吕峰军、师正杰、施以箴、杨洪涛、王润森、魏汪洋。在此一并表示感谢。

编者

2009 年 8 月于北京

目　录　Contents

第一篇　Linux 基础篇

第 1 章　Linux 概述　2

1.1　Linux 背景知识……2
1.1.1　Linux 名字的由来……2
1.1.2　自由软件库 GNU……2
1.1.3　标准程序接口 POSIX……3
1.1.4　GPL 许可证……4
1.1.5　Linux 发展史……5
1.2　Linux 功能……6
1.2.1　操作系统基本概念……6
1.2.2　Linux 基本特性……7
1.2.3　Linux 的应用范围……7
1.3　Linux 适合我吗……8
1.3.1　Linux 适合的人群……8
1.3.2　Linux 不适合的人群……9
1.4　进阶阅读材料……10
1.5　小结……10

第 2 章　Linux 安装　11

2.1　版本选择因素……11
2.1.1　包管理系统……11
2.1.2　桌面系统……14
2.1.3　选择适合自己的发行版……15
2.2　获得 Linux 安装文件……15
2.2.1　从官方网站下载光盘镜像文件……16
2.2.2　购买 CD 或 DVD……17
2.2.3　索取免费 CD……17
2.3　启动管理器……17
2.3.1　GRUB 简介……17
2.3.2　menu.lst……18
2.3.3　启动菜单的交互式操作……19

2.3.4 命令行操作……20
2.3.5 应用实例……21
2.3.6 另一款启动管理器 LILO……21
2.4 丰富的 Linux 安装方式……22
2.4.1 使用 Wubi 安装……22
2.4.2 使用光盘启动安装……23
2.4.3 从硬盘安装……24
2.5 Linux 分区与挂载点……26
2.5.1 硬盘的分区简介……26
2.5.2 Linux 分区编号的意义……26
2.5.3 Linux 系统的挂载点……26
2.6 安装过程详解……27
2.6.1 分区前的基本设置……27
2.6.2 硬盘分区……28
2.6.3 分区后的确认步骤……30
2.7 小结……31
第 3 章 桌面环境简介 32
3.1 系统登录……32
3.1.1 启动……32
3.1.2 登录……33
3.2 认识桌面环境……34
3.2.1 桌面上都是些什么……35
3.2.2 主菜单项……36
3.2.3 使用虚拟桌面……39
3.3 美化系统……41
3.3.1 更改桌面背景……41
3.3.2 美化登录方式……42
3.3.3 使用其他主题……43
3.3.4 美化系统字体……45
3.4 开启酷炫的桌面特效……46
3.4.1 安装显卡驱动……46
3.4.2 设置 3D 立方体效果……47
3.5 定制系统以适合自己的需要……48
3.5.1 把自己喜爱的软件添加到面板……48
3.5.2 定制主菜单……51
3.5.3 在桌面上显示或隐藏图标……52

3.5.4 调节笔记本电脑的屏幕亮度……53
3.6 X Window 系统……54
3.6.1 简介……54
3.6.2 发展历史……56
3.6.3 字体的安装概述……56
3.6.4 安装核心字体……57
3.6.5 安装 XFT 字体……58
3.7 小结……58
第 4 章 Linux 基本配置 59
4.1 硬件管理……59
4.1.1 硬件管理概论……59
4.1.2 列出所有 PCI 设备……59
4.1.3 查看内核信息……60
4.1.4 显示 Ubuntu 版本信息……61
4.1.5 查看内核加载的模块……61
4.1.6 查看 USB 设备……61
4.1.7 查看网卡状态……62
4.1.8 查看 CPU 信息……63
4.1.9 显示当前硬件信息……63
4.1.10 查看当前的内存使用情况……63
4.1.11 显示内核输出信息……64
4.1.12 显示当前系统中 HAL 硬件信息……65
4.1.13 相关硬件信息的目录及配置文件……65
4.2 硬件信息目录/proc……65
4.2.1 查看该目录下的文件……65
4.2.2 系统信息文件……66
4.2.3 进程相关目录……66
4.3 启动过程详解……67
4.3.1 Linux 启动过程……67
4.3.2 init 进程在启动时所做的工作……69
4.3.3 管理自启动项目……70
4.3.4 常见系统服务列表……73
4.4 个性化终端……74
4.4.1 修改终端的外观……74
4.4.2 修改终端的系统变量……75
4.4.3 设置终端提示符格式……75

4.4.4 修改终端提示符……76
4.4.5 将更改写入配置文件……76
4.5 连接到互联网……76
4.5.1 宽带拨号用户……76
4.5.2 局域网用户……79
4.5.3 双机互联共享上网……80
4.5.4 无线上网……81
4.6 中文化……82
4.6.1 中文化概述……82
4.6.2 安装中文支持……83
4.7 安装中文输入法……83
4.7.1 Fcitx 输入法……83
4.7.2 Scim 输入法……85
4.8 Locale 简介……86
4.8.1 Locale 内容详解……86
4.8.2 Locale 的作用及定义文件……87
4.8.3 设置 Locale……88
4.9 小结……88
第 5 章 文本编辑器与正则表达式 89
5.1 Ubuntu 默认的文本编辑器 gedit……89
5.1.1 程序界面……89
5.1.2 基本设置……90
5.1.3 安装插件……91
5.2 瑞士军刀 vim……92
5.2.1 初识 vim……93
5.2.2 命令模式中的操作……95
5.2.3 命令行模式可执行的命令……99
5.2.4 vim 的基本设置……100
5.2.5 安装插件概述……102
5.2.6 目录浏览器……103
5.2.7 Tag 列表……103
5.2.8 标签页浏览……104
5.3 强大的 Emacs……104
5.3.1 基本配置……105
5.3.2 设置系统配色方案……106
5.3.3 行号显示……106

5.3.4 标签页浏览……107
5.3.5 建立 C 语言开发 IDE……107
5.3.6 多内码支持……108
5.3.7 基本操作……108
5.4 正则表达式简述……111
5.5 小结……114

第二篇 Linux 系统管理篇

第 6 章 Shell 基本命令 116

6.1 Shell 简介……116
6.1.1 Shell 的种类……116
6.1.2 Shell 的作用……117
6.1.3 Shell 的启动……118
6.2 Shell 基本技巧……119
6.2.1 自动补全……119
6.2.2 使用别名……121
6.2.3 查看最近使用过的命令……123
6.2.4 调用历史记录中的命令……124
6.2.5 history 的使用技巧……125
6.2.6 键绑定……127
6.2.7 Shell 的变量……129
6.3 重定向与管道……130
6.3.1 文件描述符……130
6.3.2 重定向符的使用……130
6.3.3 管道……133
6.4 文件内容相关命令……134
6.4.1 输出文件内容……134
6.4.2 文本文件比较……138
6.4.3 文本处理……141
6.4.4 awk 文本处理程序……145
6.4.5 sed 流编辑器……148
6.4.6 grep 文本查找命令……150
6.4.7 显示可执行文件详情……151
6.5 文件操作基本命令……155
6.5.1 显示文件信息命令 ls……155

6.5.2 文件移动、复制、删除命令……157
6.5.3 创建文件链接、分割文件……159
6.5.4 目录操作……161
6.5.5 文件查找命令 find……164
6.5.6 快速查找命令及特殊查找命令……165
6.6 其他命令……167
6.7 强大的手册 man……173
6.8 本章命令列表……173
6.9 小结……175
第 7 章 文件管理 176
7.1 Linux 的目录……176
7.1.1 Linux 目录结构……176
7.1.2 Linux 的一些重要目录简介……177
7.2 文件系统格式……178
7.2.1 文件系统概述……178
7.2.2 ext2、ext3 文件系统……180
7.2.3 日志文件……181
7.3 加载其他文件系统……182
7.3.1 加载与卸载文件系统……182
7.3.2 挂载光盘镜像……183
7.3.3 挂载移动硬盘、U 盘……183
7.3.4 挂载 SMB 文件系统……184
7.3.5 挂载 NFS 文件……184
7.3.6 挂载所有文件系统……185
7.3.7 卸载文件系统……185
7.3.8 开机自动加载……185
7.4 文件管理器……187
7.4.1 文件管理器界面……187
7.4.2 打开文件……189
7.4.3 更改文件的打开方式……190
7.4.4 文件管理……191
7.4.5 新建目录……191
7.4.6 给目录、文件增加徽章……192
7.4.7 创建压缩文件……192
7.4.8 可移动存储设备管理……193
7.4.9 权限设置……193

7.5 chmod 命令设置权限……194
7.5.1 字符方式设置文件权限……194
7.5.2 数字掩码方式修改文件权限……195
7.6 刻录 CD/DVD 光盘……196
7.6.1 文件管理器刻录……196
7.6.2 专用软件刻录……196
7.7 压缩文件管理……198
7.8 小结……200

第 8 章 软件包管理 201

8.1 安装前的准备工作……201
8.1.1 源服务器简介……201
8.1.2 添加源列表……202
8.2 使用主菜单中“添加/删除”命令管理软件……205
8.2.1 “添加/删除”命令界面……205
8.2.2 安装软件……206
8.2.3 卸载软件……207
8.3 使用新立得软件管理器管理软件……207
8.3.1 新立得软件管理器界面……207
8.3.2 软件源管理……209
8.3.3 软件管理……210
8.4 安装其他格式的软件包……211
8.4.1 安装 Deb 包格式的软件……212
8.4.2 编译、安装提供源代码的软件……213
8.5 软件包管理命令……214
8.5.1 apt 系列……214
8.5.2 dpkg 系列……218
8.6 小结……220

第 9 章 磁盘管理 221

9.1 用 Live CD 进行磁盘管理……221
9.1.1 磁盘分区……221
9.1.2 格式化……222
9.1.3 更改分区大小……223
9.2 磁盘管理命令……223
9.2.1 分区命令……224
9.2.2 格式化命令……229

9.3 查看磁盘使用情况 230
9.3.1 fdisk 查看分区情况 230
9.3.2 查看/dev/disk 目录相关内容 231
9.3.3 查看/proc/partitions 中的分区信息 233
9.3.4 查看磁盘已挂载分区使用情况的命令 df 233
9.3.5 使用 du 查看目录信息 234
9.4 hdparm 查看及设置磁盘信息 235
9.5 磁盘内容复制 237
9.6 小结 238
第 10 章 进程管理 239
10.1 进程知识简介 239
10.1.1 进程的基本概念 239
10.1.2 守护进程 240
10.1.3 僵尸进程 240
10.2 进程监视 240
10.2.1 进程显示命令 ps 241
10.2.2 memstat 显示共享库的内存占用 244
10.2.3 pstree 以树型列表方式显示进程 244
10.2.4 进程筛选命令 pgrep 245
10.2.5 动态监视进程命令 top 246
10.3 终止进程 248
10.3.1 信号的基本概念 248
10.3.2 kill 命令 249
10.3.3 按进程名终止进程 pkill 249
10.4 进程高级管理 250
10.4.1 进程的后台执行 250
10.4.2 进程的优先级 252
10.4.3 进程自动调度 at、crontab 253
10.5 小结 254
第三篇 Linux 应用篇
第 11 章 网络配置管理 256
11.1 网络管理概述 256
11.2 iptables 简介 258

11.3 监视网络性能……261
11.3.1 使用 ping 检测网络性能……261
11.3.2 使用 netstat 监视网络性能……262
11.4 配置 TCP/IP……264
11.4.1 显示特定网络设备的状态……265
11.4.2 设置网络设备的 TCP/IP 参数……265
11.4.3 给网络设备取别名……265
11.5 DNS 管理简介……266
11.5.1 DNS 简介……266
11.5.2 DNS 的配置文件……267
11.5.3 nslookup 命令……269
11.5.4 dig 命令……270
11.6 路由管理简介……271
11.7 端口管理简介……272
11.8 远程登录简介……273
11.8.1 Vino 远程桌面……274
11.8.2 VNC 远程桌面……275
11.8.3 Telnet 登录……276
11.8.4 SSH 登录……277
11.9 小结……281

第 12 章 网络相关软件 282

12.1 Firefox 网络浏览器……282
12.1.1 Firefox 特色……282
12.1.2 Firefox 界面……283
12.1.3 初步设置……283
12.1.4 安装插件……286
12.2 精致小巧的 Opera 浏览器……289
12.3 安装 IE 浏览器……292
12.4 字符界面网页浏览软件……293
12.4.1 强大的字符界面浏览器 w3m……293
12.4.2 高速字符界面浏览器 lynx……295
12.5 收发电子邮件……296
12.5.1 图形界面软件 Evolution……296
12.5.2 字符界面邮件管理……298
12.6 即时通信软件……301
12.6.1 全能的 Pidgin……301

12.6.2 可拨打普通电话的 Skype……303
12.6.3 专业强大的 IRC……305
12.7 小结……307
第 13 章 多媒体软件 308
13.1 视频播放……308
13.1.1 全能电影播放器 MPlayer……308
13.1.2 电影播放器 Totem……311
13.2 音频播放……313
13.2.1 类似 Winamp 的播放器 Audacious……313
13.2.2 类似 iTune 的播放器 Rhythmbox……315
13.3 视频处理……316
13.3.1 视频编辑软件 LIVES……316
13.3.2 视频格式转换 Avidemux……319
13.3.3 编译增强版的 FFmpeg……321
13.3.4 使用 FFmpeg 转换视频文件……322
13.4 小结……324
第 14 章 图形图像软件 325
14.1 图片浏览软件 gThumb……325
14.1.1 软件简介……325
14.1.2 图片浏览……326
14.1.3 图片处理……326
14.1.4 制作网络相册……328
14.2 系统默认的图片浏览软件 Eye of GNOME……329
14.3 图片管理软件 Picasa……330
14.3.1 功能简介……330
14.3.2 安装……330
14.3.3 设置……331
14.3.4 使用简介……332
14.3.5 网络相关功能……334
14.3.6 其他功能……336
14.4 系统默认的图片管理器 F-Spot……337
14.5 图像处理软件 GIMP……339
14.5.1 简介……339
14.5.2 界面简介……340
14.5.3 安装插件……341

14.5.4 示例……342
14.6 矢量图绘制软件……345
14.6.1 OOo 办公套件中的绘图软件 OpenOffice.org Drawing……345
14.6.2 矢量图绘制软件 Inkscape……346
14.7 CAD 软件 QCad……349
14.7.1 QCad 简介……349
14.7.2 绘制零件图……350
14.8 小结……353
第 15 章 办公软件 354
15.1 办公软件 OpenOffice.org 简介……354
15.2 OOo 文字处理……355
15.2.1 界面简介……355
15.2.2 文档排版……356
15.2.3 文本项目编号……359
15.2.4 图片与文档的混合排版……360
15.2.5 绘制简单的表格……363
15.3 OOo 电子表格……364
15.3.1 软件界面……364
15.3.2 制作复杂格式的电子表格……365
15.3.3 调整显示格式……366
15.3.4 绘制表格线……367
15.3.5 导入文本文件……368
15.3.6 图表制作……369
15.3.7 公式计算……371
15.4 OOo 演示文稿……373
15.5 IBM Lotus Symphony 办公套件……376
15.5.1 下载安装……376
15.5.2 使用简介……377
15.6 永中 Office 办公套件……378
15.7 文书浏览……380
15.7.1 正确显示中文 PDF 文件……380
15.7.2 打开 CHM 文件……380
15.8 小结……381
第 16 章 专业排版软件 LaTeX 382

16.1 LaTeX 的特点 382
16.2 安装 LaTeX 程序 383
16.2.1 安装核心程序 383
16.2.2 安装中文支持 384
16.2.3 安装 vim 的 LaTeX 插件 387
16.2.4 安装 Emacs 的 LaTeX 插件 390
16.3 基础知识 391
16.3.1 TeX 基本文件结构 391
16.3.2 关于 TeX 文件的预备知识 392
16.4 LaTeX 的文档结构 393
16.5 LaTeX 的公式排版 395
16.6 在 LaTeX 中插入图片 400
16.7 小结 402

第四篇 Linux 开发篇

第 17 章 Linux 环境中 C/C++编程简介 404

17.1 GCC 简介 404
17.1.1 GCC 功能简介 404
17.1.2 第一个 C 程序 405
17.1.3 GCC 的具体编译过程 406
17.1.4 GCC 高级技巧 408
17.2 makefile 简介 411
17.2.1 第一个 makefile 文件 411
17.2.2 对简单 makefile 文件的扩充 413
17.2.3 在 makefile 中使用函数 414
17.2.4 清理 make 过程中的中间文件 415
17.3 程序调试 416
17.3.1 GCC 编译的几种错误信息及其解决方法 416
17.3.2 GDB 简介 417
17.3.3 GDB 例程 418
17.4 C/C++开发环境 421
17.4.1 使用 vim 421
17.4.2 使用 Eclipse 424
17.5 小结 428

第 18 章 Shell 编程 429

18.1 Shell 脚本简介……429
18.1.1 特点……429
18.1.2 Shell 脚本程序的基本结构……430
18.1.3 Shell 脚本程序的几种特殊字符……431
18.1.4 用户交互……432
18.2 变量与数组……434
18.3 算术操作符……438
18.4 条件判断语句……439
18.4.1 if 语句……440
18.4.2 case 语句……441
18.4.3 select 语句……442
18.4.4 各种测试操作……443
18.5 循环语句……444
18.5.1 for 语句……444
18.5.2 while 语句……446
18.5.3 until 语句……447
18.5.4 循环语句的一些控制语句……447
18.6 使用函数……449
18.7 脚本调试……450
18.8 小结……452
第 19 章 搭建 LAMP 服务器 453
19.1 安装 LAMP……453
19.1.1 安装……453
19.1.2 测试……454
19.2 配置 Apache……455
19.2.1 配置文件……455
19.2.2 增加一个基于域名的虚拟主机……457
19.2.3 使用 SSL 加密网页传输……460
19.3 MySQL 基本操作……463
19.3.1 连接数据库服务器……463
19.3.2 MySQL 命令简介……463
19.4 PHP 基本设置……466
19.5 安装 Web 应用程序……467
19.5.1 安装 MediaWiki 维基百科……467
19.5.2 安装 WordPress 博客程序……470
19.6 小结……473

第 20 章　FTP 服务器的搭建　474

20.1　FTP 服务器安装……474
20.1.1　FTP 简介……474
20.1.2　FTP 服务器软件安装……475
20.2　图形界面软件 FileZilla……476
20.3　字符界面软件 lftp……478
20.3.1　FTP 登录……478
20.3.2　FTP 服务器目录与文件操作……479
20.3.3　下载、上传文件……480
20.3.4　镜像命令 mirror……481
20.3.5　lftp 使用技巧……482
20.4　WU-FTP 的设置……484
20.5　添加只能进行 FTP 操作的账号……487
20.6　小结……488

第五篇　Linux 安全篇

第 21 章　用户与组管理　490

21.1　用户账号的基本操作……490
21.1.1　用户账号类型……490
21.1.2　建立用户账号……491
21.1.3　批量添加用户……493
21.1.4　编辑用户账号属性……494
21.2　管理用户密码……495
21.3　组的管理……496
21.4　用户、组的配置文件……498
21.4.1　账户信息配置文件/etc/passwd……498
21.4.2　账户的影子口令文件/etc/shadow……499
21.4.3　组记录文件……500
21.5　用户、组相关命令……501
21.6　小结……503

第 22 章　系统安全　504

22.1　Linux 的良好安全系统……504

22.2 FireStarter 防火墙软件……505
22.2.1 安装、使用向导进行设置……505
22.2.2 基本设置……506
22.2.3 策略设置……507
22.3 PAM 机制……510
22.3.1 PAM 原理……510
22.3.2 PAM 配置文件……511
22.4 使用 seahorse 进行文件加密……512
22.4.1 创建密钥……513
22.4.2 文件加密与数字签名……513
22.4.3 添加数字签名……514
22.4.4 公钥管理……514
22.5 了解与分析日志文件……515
22.5.1 解读/etc/syslog.conf 配置文件……515
22.5.2 syslogd 守护进程的参数……517
22.5.3 查看日志文件……518
22.6 小结……520

第 一 篇

Linux 基础篇

- 第 1 章　Linux 概述
- 第 2 章　Linux 安装
- 第 3 章　桌面环境简介
- 第 4 章　Linux 基本配置
- 第 5 章　文本编辑器与正则表达式

第 1 章　Linux 概述

Linux 是一款与 Windows 同样优秀的计算机操作系统，可以用其来高效完成各项工作。更重要的是，Linux 是一款完全开源免费的操作系统，意思是不需要支付任何版权费就可自由使用。Linux 这一名词，在两三年前似乎还只是那些头发蓬乱，戴着眼镜的技术极客们的宠儿。但随着 Linux 的发展，使得其不光占领了大部分的服务器市场份额，还逐渐被普通用户所认可，越来越多地出现在普通用户的计算机里。本章将简单介绍一些 Linux 的背景知识。

1.1　Linux 背景知识

作为一名 Linux 用户，对于 Linux 相关名词和历史如果不能如数家珍，那就太不可思议了。本节会介绍 GNU、POSIX 标准等相关名词。这些名词在本书后续章节也会提及，所以应该对其有初步的了解。本节还会介绍 Linux 从蹒跚学步到光彩夺目的全部过程。

1.1.1　Linux 名字的由来

据 Linux 的作者 Linus Torvalds 在其自传《Just For Fun》一书中所说，在一开始并不想用 Linux。因为这个名字听上去像是 Linus 的 UNIX，这样好像有点自负。在开始开发时，Linus 准备的名字是 Freax（Freax 的英文含义是怪诞的、怪物、异想天开等）。而其 FTP 的管理员 Ari Lemke 坚持认为这个名字很不好，于是用他自己的方式将内核代码放到 FTP 站点上，并坚持要用 Linux 这个名字，所以该操作系统内核就以 Linux 这个名字被广为传播。

Linux 的吉祥物是一只名字叫做 Tux 的企鹅，如图 1.1 所示。1994 年发表 Linux 正式核心 1.0 的时候，开发者们建议 Linus Torvalds 为 Linux 内核选择一个吉祥物。Torvalds 的妻子想到的吉祥物是企鹅，因为 Linus 喜欢企鹅，小时候去动物园时还被一只企鹅咬了一口！于是就建议在想不到其他吉祥物的情况下，干脆就以这只企鹅来当吉祥物！这个建议得到大家的赞同。而且为了特别起见，这只名为 Tux 的企鹅的样子还是很有特色的。仔细观察就会发现其喙与璞均为黄色，而不是常见的白色。

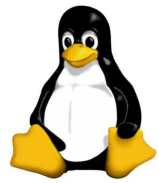


图 1.1　Linux 的吉祥物 Tux

Linux 的注册商标理所当然属于核心技术的掌握者 Linus Torvalds 所有。不过，律师 William R. Della Croce 可不这么认为；1996 年他声明 Linux 商标属于自己所有，各大发行版需向自己支付版权费。他的商业意识显然用错了地方，在各方努力下，他与 Linus 达成了庭外和解，Linux 的版权重新分配给了 Linus，Linus 也重申了 Linux 将永远开源的承诺。

1.1.2　自由软件库 GNU

GNU 这个名字是 GNU's Not UNIX（GNU 不是 UNIX）的递归缩写。这是 GNU 成立时为

了表示与商业 UNIX 系统的差别，而特别采用的名称。GNU 发音为 Guh-NOO（作为单词的 gnu 其意思是角马，发音与单词 new 相同）。

GNU 计划（其标志如图 1.2 所示）是由 Richard Stallman 在 1983 年 9 月 27 日发起的，其目的是建立一个与 UNIX 兼容、软件丰富，且可自由使用的软件库（包括操作系统内核）。Richard Stallman 最早是在 net.unix-wizards 新闻组上公布该消息，并附带一份《GNU 宣言》解释为何发起该计划的文章，其中一个理由就是要“重现当年软件界合作互助的团结精神”。



图 1.2　GNU 的标志

GNU 的软件与 UNIX 系统兼容，因为两者均符合 POSIX 标准。最初 GNU 的软件是参照 UNIX 来编写的，所以 GNU 也可看做是商业 UNIX 软件的一个自由实现，一般的 UNIX 软件在 GNU 中都有一个对应的同名软件。

由于 GNU 软件遵循 GPL 协议，所以发布时须同时提供源代码。这一方面保证了软件的衍生版本与原始版本的兼容性（UNIX 没有开源，所以在最初的代码衍生的几个 UNIX 商业版本之间出现许多重大的不同）；另一方面顶尖的软件高手不断对代码进行修改，所以使得 GNU 软件的功能不断增强。另外，由于 GNU 软件的质量比之前 UNIX 的原始版本软件还要好，后来许多 UNIX 系统上也安装了 GNU 软件；同样由于 GNU 软件性能优良，还被广泛地移植到 Windows 和 Mac OS 等各种操作系统上。

GNU 是个软件的组合，其不仅是鼓励软件开发者开发 GNU 的软件，还大量吸纳着其他优秀的自由软件，如 TeX 和 X Window。所以 GNU 软件资源极为丰富且功能强大。GNU 拥有大量的应用软件，但 GNU 的目标是创建一个完整的操作系统，而当时缺少的就是一个自由的操作系统内核。Linux 的出现弥补了这个空白。然后 Linux 与 GNU 丰富的软件结合形成了极其旺盛的生命力，发展极为迅速，也达成了 GNU 最初的理想：实现一个完全自由的操作系统。

所以采用 Linux 内核，加上 GNU 软件库中软件的操作系统应该被称为 GNU/Linux。或许 GNU 觉得不能采用自己的内核是个遗憾，所以 GNU 自己的内核也在开发中；该内核名为 Hurd，目前已发布了三个版本。

1.1.3　标准程序接口 POSIX

POSIX（Portable Operating System Interface for Computing Systems）是由 IEEE 和 ISO/IEC 开发的一个标准。该标准描述了操作系统的调用服务接口，用于确保符合该标准的应用程序可以在多种操作系统上移植运行。

由于早期各大 UNIX 系统之间的软件和系统调用并不兼容，1980 年一个 UNIX 用户组（usr/group）试图将 AT&T 的系统 V 和 Berkeley 的 BSD 系统的调用接口统一，并于 1984 年产生了/usr/group 标准，这就是 POSIX 标准的前身。1986 年 4 月 IEEE 公布了试用标准，在 1988 年 9 月第一个正式标准（IEEE 1003.1-1988）被批准，该标准即是 POSIX.1 标准。

1989 年 POSIX 的工作被转移至 ISO/IEC 社团，以将其制定成 ISO 标准。1990 年 POSIX.1 与已经通过的 C 语言标准联合，正式批准为 IEEE 1003.1-1990（也是 ANSI 标准）和 ISO/IEC9945-1:1990 标准。

POSIX.1 标准仅包括了基本的系统服务，但仅包括这些内容还不足以保证程序的可移植性。因此 1990 年 IEEE POSIX 的其他标准制定工作也开始进行，计有命令与工具标准（POSIX.2）、

测试方法标准（POSIX.3）、实时 API（POSIX.4）等，这些标准涵盖了计算机系统的各个方面。

POSIX 对于 Linux 来说，有着重要的意义。因为 Linux 从一开始就遵循该开放标准，这保证了 Linux 与其他 UNIX 系统的兼容。在 0.01 版的 Linux 源代码/include/unistd.h 文件中就已经定义了几个有关 POSXI 标准要求的常数符号，并且在注释中写道：“OK，这也许是个玩笑，但我正在着手研究它呢。”

1.1.4 GPL 许可证

GPL 是 GNU General Public License（GNU 通用公共许可证）的缩写形式，其要求软件发布时必须同时发布源代码，并允许任何用户能够以源代码的形式将软件复制或发布给别的用户。如果一个软件使用了遵循 GPL 许可证的任何软件的一部分，那么该软件也必须继承 GPL 许可，因此而成为 GPL 软件，也就是说必须随该软件一起发布源代码。

基于 GPL 的软件允许商业化销售，但必须再开放源代码，这个特点多少限制了软件的商业价值。因为有的安全软件如果公布了源代码会大大降低其安全性，有的专利算法也会由于公布源代码而失去商业价值。因此对 GPL 持批评态度的人也大有人在，他们认为“GPL 是一种病毒，因为它具有太强的传染性了”。

一般来说，GPL 具有如下的特性。

- GPL 授予软件用户（程序接受人）以下权利，或称“自由”：以任何目的运行此程序的自由；以学习程序工作机理为目的，对程序进行修改的自由；再发行复制件的自由；改进此程序，并公开发布改进的自由。
- GPL 只在程序再发布时发生效力。如果修改后只供自己的使用，不发行的话对软件的修改可以不开放源代码。而且该许可只对软件本身有效力，而对软件的输出并无效力（除非输出的是软件本身）。如用户张三使用开源软件书写的文档，用 GIMP 处理的图片，用 BLENDER 建立的 3D 模型，这些文件著作权属于张三，而不是属于 GIMP 或 blender 的开发者。
- GPL 并不是免费的代名词。GNU 的创建人就是凭借销售其开发的 Emac 软件获得较为丰厚的收入后，才有充足的精力和财力创建 GNU 组织的。GPL 并不排斥对自由软件进行商业性质的包装和发行，也不限制在自由软件的基础上打包发行其他非自由软件。任何接受 GPL 许可证的接受人都有修改、复制、再发行作品或作品的演绎版本的授权。许可证接受人可以对此项服务收取费用，反之亦然。这一点是 GPL 与其他禁止商业用途的自由软件许可证的最大不同。Stallman 认为自由软件不应限制其商业用途，同时 GPL 也清楚地说明了这一点。
- 遵循 GPL 许可的软件不一定就可以任意传播，因为有些软件有其正式的版权，GPL 在发布软件或者复制软件时声明了限制条件。但是，从用户的角度来看，这些条件对于自由使用并没有什么影响。
- GPL 许可证文本的版权属于自由软件基金会。但遵循 GPL 许可的软件著作权属于作者所有，只有著作权所有者才有权利对于违反许可证的行为提起诉讼。
- 尽管 Linux 内核也属于 GPL 范畴，但 GPL 并不适用于通过系统调用而使用内核服务的应用程序。通常把这种应用程序看做是内核的正常使用，同样，一个软件对其他 GPL 程序接口进行调用，可以认为并未使用 GPL 软件，就不必受 GPL 许可证的约束。

Copyright 规定了所有者的权益，而 GPL 有 Copyleft 条款，意思是遵守该协议的著作人将

放弃版权。对于使用者而言，只要使用时遵守 GPL，就可以无视 Copyright。如果使用者不遵循 GPL 协议，那么 Copyright 就发生作用，著作权所有人可以对侵权行为发起诉讼。在美国已经有人打赢了类似的官司。



在 Linux 发展过程中，本节所介绍的几个名词所代表的意义对于 Linux 来说有着极其重要的作用。GNU 发展出了丰富的软件，Linux 提供了系统内核，相得益彰，于是 GNU 与 Linux 的发展走上了快车道。而 POSIX 标准的提出无疑给 Linux 一个和当时主流的计算机世界接轨的机会，让符合 POSIX 标准的程序可以在 Linux 中运行无疑大大提高了 Linux 系统的生存能力和影响力。GPL 则为 GNU 软件的发展立下了汗马功劳，其使得 GNU 软件具有开放性与一致性，也提高了 GNU 软件的质量。

1.1.5　Linux 发展史

Linux 现在的发展极其迅速，其拥有许多发行版，桌面市场占有率也在逐步上升，服务器市场更是占据了大部分的市场份额。这些成绩离不开那些早期的 Linux 的缔造者和爱好者，他们一直为 Linux 的发展做着无私的奉献，我们不应该忘记他们和一些有纪念意义的历史时刻。Linux 的迅速发展得益于互联网、GNU，其基础则是 UNIX 和 MINIX，所以，在提及 Linux 历史时不得不提及这些要素。下面是 Linux 及其相关技术的历史介绍。

1987 年，由于当时强大的 UNIX 采用商业发布模式，不允许发布源代码。这种情况对于学习操作系统的学生来说是一个很大的困难，因为其无法接触到操作系统的核心实现。Andrew Tanenbaum 教授有感于此，从 1984 年开始编写与 UNIX 功能类似，并完全兼容的 MINIX。这是一个多平台的 mini UNIX 版本，所以称为 MINIX，其在发布时带有完整的源代码。

1991 年 8 月 25 日，芬兰赫尔辛基大学的一名学生 Linus Torvalds 在 comp.os.minix 张贴了如下内容：

“使用 MINIX 的用户大家好：

我正在写一个（免费的）基于 386（486）机器的操作系统（只是个人爱好，代码不会太大或专业到像 GNU）。这个系统从 4 月开始酝酿，现在已经准备好开始写了。我希望 MINIX 用户给我喜欢或不喜欢的 MINIX 的特性，我的操作系统和 MINIX 在许多特性上（采用 MINIX 文件系统）很相像…欢迎任何建议，虽然我不一定会采纳。

Linus(torvalds@kruuna.helsinki.fi)

附注：当然，这个操作系统没有使用一行 MINIX 源码，拥有多线程的文件系统。由于使用了 386 的任务切换特性，所以该操作系统没有可移植性，并且只能使用 AT 硬盘。”

这段话的背景是：当时 MINIX 用户对于 MINIX 有许多的建议与意见，而 MINIX 的开发者 Tanenbaum 教授教学任务繁重，无法及时响应这些建议。所以才有了 Linus 说希望 MINIX 的用户给他建议。

1991 年 10 月 5 日，Linus 在 comp.os.minix 新闻组上发布消息，正式向外宣布 Linux 内核系统的诞生（Free minix-like kernel sources for 386-AT）。这段消息可以称为 Linux 的诞生宣言，并且一直广为流传，因此 10 月 5 日可以称得上是 Linux 的生日。该版本包括简单的磁盘驱动程序和一个文件系统，可以运行 bash 和 gcc 编译器，但该系统并没有良好的用户界面以及文档。在此之后，该系统引起了各地 hack 的注意，并以每两个星期出一次修正版的速度成长。

1994 年 3 月 14 日，第一个正式版本 1.0 发布；而 Linux 的讨论区也从 comp.os.minix 中独

立出来成为 alt.os.Linux，而后又更名为 comp.os.Linux。后来由于访问者数量越来越多，又将此区分为许多组 comp.os.Linux.*。

同年，Marc Ewing 建立 Red Hat Linux，用以解决 Linux 易用性方面的问题。Red Hat 发行版包含 Linux 内核、第三方软件、文档，以及初级技术支持等，售价为 50 美元，由于其解决了 Linux 的易用性的问题，Red Hat 迅速成为领先的 Linux 发行版。

1996 年 2 月 9 日，Linux 2.0 版发布。同年，美国国家标准技术局的计算机系统实验室确认 Linux 1.2.13 版（由 Open Linux 公司打包）符合 POSIX 标准。

1999 年 1 月 26 日，Linux 2.2 发布；GNOME 1.0 发布；支持 Linux 2.2 的 Red Hat 6.0 发布；IBM 推出全面支持 Linux 计划；HP 宣布支持 Linux。由此可见，这些大公司在十年前已经纷纷宣布支持 Linux 了。与此同时，各软件巨头也提供了软件的 Linux 接口；这样，Linux 的用户也就可以调用软件了。这些现象的出现，说明 Linux 的影响已经不局限于技术极客，在商业领域也开始拥有了足够的影响力。

现在，Linux 已经拥有 Red Hat、open SUSE、Ubuntu 等许多大的发行版，小的发行版有数百种。同时，其在其他领域如嵌入式领域也有了很大的发展，内核也已发布到 2.6.24 版。

1.2 Linux 功能

现在，对 Linux 相关的名词及历史有了初步的了解，但这还远远不够。使用 Linux 最应该知道的是 Linux 具有什么功能，是否能满足工作生活的需要。

1.2.1 操作系统基本概念

Linux 是一种操作系统，或者按其狭义观点来说是一种操作系统的内核，搭配上其他软件如 GNU 软件后就可以提供实用的功能。现代生活已经离不开计算机，但什么是操作系统，大多数人却只知道一个模模糊糊的概念。“操作系统，就是 Windows 吧。”如此的概念，甚嚣尘上。但其实，Windows 是操作系统，但也是林林总总的操作系统中的一个重要成员而已。在网络服务器市场，Windows 的市场占有率其实还很低的。

为了理解操作系统的概念，还需要知道一些计算机的基本常识。从逻辑概念上来说，计算机由软件和硬件两部分组成。硬件最直观的解释就是关机时也可以看见的部分，如主机、显示器、鼠标、键盘之类，其中最重要的是主机箱里的东西，里面包含着完成计算机核心功能所需要的部件，如 CPU（计算机的大脑）、内存、硬盘（计算机的记忆）、主板（计算机的身体，其他东西都连接在主板上）。

软件是计算机完成某些工作而编写的可运行代码。软件从系统的观点来看，由操作系统和应用软件这两部分组成。

1．操作系统

操作系统直接管理硬件，完成一些系统底层的功能，如进程管理（管理应用程序如何运行）、文件管理（管理系统中所有设备和文件）、存储器管理（提供读、写文件的方法），然后给应用程序提供一些接口，供其使用。目前个人计算机上使用的操作系统，主要有 Windows 和类 UNIX 系统（如 Linux）。Windows 系统在个人电脑市场占据了绝大多数市场，而 Linux 在服务器市场是绝对的主流，现在 Linux 也在桌面系统发力，最近几年发展速度很快。

2．应用软件

应用软件是运行在操作系统上的一些软件。这些软件可以完成特定的功能，如办公、影音播放、网络浏览等功能。一般来说，应用软件只能运行在特定的平台上，如 Mircosoft Office 只能运行在 Windows 系列的操作系统之上，而 Deb 包文件只能被 Linux 系统识别等。

1.2.2　Linux 基本特性

应用软件运行于操作系统之上，一方面与系统交互，获取操作系统提供的功能，另一方面提供直观的人机操作界面，供用户使用。计算机用户经常使用的文字处理软件、图像处理软件、电子表格软件等就属于应用软件。一般来说，Windows 和类 UNIX 的软件是不能共用的，而类 UNIX 系统，由于其符合 POSIX 标准，可以互相通用。与 Windows 相比，Linux 具有如下特性。

- 硬件需求低：不会动辄要求 2GB 内存，3.0GB 以上 CPU。在比较原始的 PIII866 或更低的老机型上也可以流畅地运行。就可以让旧机型发挥余热，如充当网关或 FTP 服务器等。
- 架构开放、跨平台：Linux 是类 UNIX 系统，其遵循世界标准规范 POSIX 1003.1，该标准定义了一个 UNIX 类操作的最小界面，Linux 从 1.2 版起完全支持该标准。凡是支持该协议的软件，都可以不加修改地在 Linux 上运行，如 SCO UNIX 中的软件就可以直接在 Linux 中运行。
- 系统安全性强：由于系统架构的原因，普通用户拥有的权限很少，所以即使感染病毒，也基本不可能对系统文件成影响，而高级用户具备了较高的技术，可以防范恶意代码的运行。因此 Linux 具有极高的安全性，很少有病毒在 Linux 平台大规模流行。
- 系统稳定性高：Linux 开放源码，在各大开源公司及全世界各地顶尖 Hacker 的努力下，不断解决系统可能存在的问题，同时一直对 Linux 内核代码进行优化、升级，因此 Linux 稳定性强。
- 完全免费：由于 Linux 遵循 GPL 协议，所以会免费提供码供用户或大的 Linux 发行公司使用。

1.2.3　Linux 的应用范围

在简单了解了什么是操作系统后，就可以回答“Linux 能做什么”的问题了。简单地说，在 Windows 能完成的工作，在 Linux 里同样可以完成；而且 Linux 具有以上所述优点，所以可以在很多方面有着丰富的应用，下面列出一些 Linux 实际应用。

- 办公软件：Linux 提供了功能强大并且与 Micorsoft Office 兼容的办公套件，可以提供文字处理、电子表格、幻灯片制作等相关功能。
- 多媒体方面：Linux 可以进行音视频播放、音视频处理、图像处理、3D 建模等相关多媒体方面的操作。
- 网络应用方面：Linux 可以让用户方便地浏览网页、收发电子邮件等。
- 编程：在 Linux 下一般默认安装 gcc、Python 等程序编译器，使得学习编程更加容易。文本编辑器加独立的编译器可以让使用者更清楚地知道程序编制的各个过程，不会像只会使用 IDE 的人，离开编译器就什么也做不了。
- 嵌入式应用：这算得上是 Linux 最具发展潜力的领域。从技术上看，Linux 操作系统具有内核稳定、可靠性高、实时性好、可裁剪、内核小等多种优点。因此，Linux 在工控设备、手持设备、信息家电等领域具有广泛的发展前景。除此之外，各类智能控制设备、

智能仪表、自动控制设备等，都可以利用嵌入式 Linux 作为其内部操作平台。据有关技术人员介绍，通常情况下，工控设备采用的操作系统为 DOS 系统或专用系统，前者的问题是单应用、单用户，很难支持新的智能设备；后者通常是封闭系统，缺少灵活性，而且一旦出了问题，维护费用很高。与这些系统相比，嵌入式 Linux 的开放性、稳定性以及对多任务、多用户的支持都很值得称道。

- WWW 服务：Linux 被广泛应用于在互联网中提供 WWW 服务。据统计 Linux 服务器在全球占据了将近 60%～70%的份额，用 Linux 提供 WWW 服务的软件组合叫 LAMP。Linux 还被广泛用作 DNS 服务器、NIS 服务器、NFS 服务器等。
- 文件及打印服务：Linux 下的 Samba 服务器，可以轻松地向用户提供文件及打印服务，还可以提供更高级的功能，如磁盘定额管理等。
- 数据库服务：数据库在服务器市场是一个绝对重要的应用。Linux 提供了对数据库系统的全力支持，各大数据库厂商纷纷看好 Linux 广阔的前景，先后推出基于 Linux 系统下的大型数据库。目前可以在 Linux 运行的大型数据库系统有：Sybase、Oracle、IBM DB2、MySQL 等。
- 网络服务：例如将 Linux 系统用作路由器等，这些系统通常情况下只安装简单的网络服务功能软件包。大多情况下是将一台配置很低的机器插两块网卡，用作一个简易的路由器或网桥。对于安全性要求较高的网络，还可以将 Linux 机用作防火墙（IP Chain）来使用，以杜绝外部入侵者的破坏。另外，Linux 还可以毫不费力地用作代理服务器（Socket、Squid 等），也可以实现主机接入网络，局域网的其他电脑通过该主机连接到互联网中。

1.3 Linux 适合我吗

近几年来，关于 Linux 的原教旨主义者和 Windows 的爱好者之间的论争就没有停息过。支持 Linux 的人认为 Linux 功能强大，使用自由，Windows 只不过是一个没有技术，只会在第三世界国家骗人的系统。而支持 Windows 的人则认为，Linux 使用难上手，对多媒体的支持不够，不能打游戏，只不过是一群技术极客的玩物，而不适合于普通用户使用。

1.3.1 Linux 适合的人群

其实，操作系统只是一个工具，重要的是要选择适合自己的系统，而不是参与到这些没有什么意义的争论中去。一般来说，Linux 可能适合下列人群。

- 追求时尚的用户：Linux 的 3D 桌面特效设置得当，可以大大提高系统的美观程度。对于追求时尚特色的用户来说，是一个不错的选择选择，这也是很多人选择 Linux 的第一个原因。
- 计算机专业学生：因为 Linux 开放了底层的设置。通过 Linux 的使用，可以清晰地理解操作系统的运行机制。
- 进行科学研究，与国外进行较多交流的用户：与国外进行科研方面或技术方面的交流，就可以使用 Linux。举一个简单的例子，“自然”和“科学”杂志对投稿的要求是采用 LaTeX，许多出版社要求的稿件也必须是 LaTeX 的，因为 LaTeX 版本格式严谨，排版美观，绝对是 Word 之类的办公文字处理软件所不能比拟的。
- 喜欢自己掌控系统的用户：Linux 可以自行设置系统的一切方面，可以修改内核，取消

对自己无用的模块。根据自己的机器进行优化，然后编译出一个适合自己系统的内核，以获得最好的性能。也可以对系统中各选项进行设置，控制程序和服务的行为。这一切都是闭源的商业系统不可想象的。

- 喜欢自己解决问题的用户：因为 Linux 提供了无限的可能来使用机器，所以也提供了几乎无限的会出麻烦的途径。普通的 Linux 没有商业支持，问题必须自己提交到相关的论坛，由网友义务提供帮助，这也是 Linux 最被人诟病的地方。但选择了 Linux 就必须接受这样的现实，而且在自己解决问题的过程中，会对系统有更深的了解，同时也可能会产生使用 Linux 的另一个奇妙想法。
- 普通上班族：Linux 提供的办公软件和网络浏览、多媒体播放软件等一般可满足绝大多数情况下的使用要求。
- 对安全性能有较高要求的用户：前面介绍 Linux 相关特性时也提到过，由于 Linux 系统设计的原因，安全性能很高，以及软件之间的不兼容，所以 Windows 中猖獗的病毒到了 Linux 平台就完全失去了威胁。
- 要求获得最高使用效率的用户：Linux 更多采用命令行和快捷键的操作，对于文本的操作有许多命令，而且使用最多的 vim 与 Emacs 均有极其繁多的快捷键，熟练使用快捷键可以极大地提高处理文件的效率。Shell 对于脚本文件的支持也是极好的，对于重复或有规律性的操作通过简单的 Shell 编程更是让 Linux 的使用如虎添翼。
- 版权意识浓厚的用户：Linux 遵循的是号称为“Copyleft”的 GPL 许可制度，所以使用不存在版权问题，就算是 Redhat 的企业版，也是可以自由下载使用的。使用 Linux 可以避开盗版这个道德陷阱。
- 业界顶尖的影视 3D 制作公司如皮克斯（制作了许多优秀的动画片，如料理鼠王等）使用的也是集群式的 Linux 系统，因为 Windows 不具有这方面的实力。

1.3.2　Linux 不适合的人群

当然，西谚云“此之美味佳肴，彼之穿肠毒药”。所以 Linux 在适用于一些人的同时，对另一些人来说也无异于是一个外星怪兽，避之唯恐不及，是不可能去使用与爱上 Linux 的。这些人可能包括以下几种。

- 喜欢完全傻瓜式系统的用户：Linux 提供了强大的设置系统的功能，一方面可以提高系统性能性能，但另一方面也对还不懂得怎么设置系统的用户造成了很大的困惑。Linux 从入门到精通会遇到很多的问题和困难，如果没有恒心，又静不下心了处理问题，那还是别趟这个浑水，用 Windows 吧。
- 完全沉浸在游戏中的用户：相对来说，Linux 支持的游戏真是比较少的。而即使通过 Wine 勉强运行起来的游戏，性能肯定是比不了在 Windows 中的效果。如果要打游戏，最好的选择仍然是 Windows。
- 工作中必须使用只能在 Windows 平台运行的行业软件的用户：从事 Windows 平台软件的开发人员明显是不适合使用 Linux 系统的，当然这些人中包括比尔·盖茨先生，这些人注定是不适合 Linux 的。
- 对版权不敏感的用户：Windows 是商业发行模式，所以请不要使用“出于学习的目的使用是不违反版权法的”说法来自欺欺人了。使用没有版权的 Windows 操作系统就是在使

用盗版。当然，可能许多人对版权之类的东西不敏感，经常告诉自己之所以不用盗版的原因是正版软件太贵。但其实也很少有人购买过共享软件，即使这些软件一般注册费用为二三十元。许多人在用软件时，首先想到的事情就是先找破解或注册码，而不是去寻找自由且强大的类似产品。例如，许多人一提起数学软件就是 Mathamatica、MatLab，而不理会同样强大且可自由使用的 Scilab、Maxima；要用图像处理软件就肯定是 Photoshop，而根本没有想到 GIMP 这个强大免费的软件。

1.4 进阶阅读材料

要知道 Linus 是如何开创这个美妙的系统并如何将其发扬光大，请参见他的自传《Just For Fun》；从这本书中，可以领略 Linux 的基本文化。也可以看到 Linus 是如何从一个几个月足不出户的书呆子，成为一个庞大的开源系统的优秀领袖的过程。

要完全了解 Linux 系统，最好的方法是阅读源代码。像 Linus 所说的“Read the Fuck Code”。Linux 的源代码可由 Linux 官方网站获得，地址是 http://www.linux.org。不过要注意的是，对最新版本 Linux Kernel 的简单分析显示，源代码数量超过了 1 千万行，除了注释与空白行之外的代码超过 600 万行，其中绝大多数是 C 语言代码。可以在源代码的顶级目录中使用“find -name "*.[chS]" | xargs cat | wc -l”来获得源代码的行数。阅读代码一方面可以深入学习领略 Linux 内核精妙的设计，另一方面如果对于 Linux 的理解足够深入的话，也可以顺便修正一下代码中的 Bug，并将其提交给 Linux.org。

阅读源代码，可由这本书入手《Linux 0.11 内核详细注释》，该书作者是赵炯博士。这本书以第一个功能比较完善的 0.11 版内核源码为例分析了 Linux 内核的实现。选择 0.11 版的理由是其代码结构已经类似于现代版本，而且总代码只有 1 万多行，要通读比较容易。赵炯博士还在其个人网站上提供了该书 PDF 版的免费下载，有兴趣的读者可以去如下网址 http://oldlinux.org/index_cn.html 下载。

当然，0.11 版的内核缺少许多现代操作系统的特性，如果一定要从最新的 2.x 版内核看起，可以参考如下书目，当然要阅读这些比较深入的书籍，扎实的 C 语言功底是必不可少的。

- 《Linux Device Drivers，2nd Edition》；
- 《Understanding the Linux Kernel，2nd Edition》；
- 《Linux 内核源代码情景分析》；
- 《Linux 内核 2.4 版源代码分析大全》；
- 《Linux Kernel Development》。

1.5 小结

本章介绍了 Linux 操作系统的基本概念，包括 Linux 的一些基本名词及其主要功能的简单介绍。本章是一个入门指引，并无深奥难懂的内容。不过，为了理解 Linux 操作系统与 Windows 操作系统在文化上的差异，可以先阅读 1.1.1 节内容以了解 Linux 的基础概念。要了解 Linux 文化，最好的方法是阅读 Linux 创始人 Linus 的自传《Just For Fun》，这是一本很有意思的书，文笔不错、内容翔实，详细回忆了 Linux 产生的过程及发展的历程，是所有的 Linux 用户都应该阅读的好书。如果要深入内核，则可以参考 1.4 节内容阅读有关内核分析的书籍。

第 2 章　Linux 安装

第 1 章介绍了 Linux 系统的历史及各种特色，读者应该对 Linux 有了初步的了解。但是“纸上得来终觉浅，绝知此事须躬行”，要真正了解 Linux 的特性，还是应该自己安装一个 Linux 系统，用其完成各项工作，才会有切身的体会。为了帮助没有经验的用户选择适当的发行版，本章介绍了相关发行版的特色，供用户参考。当然，本章的主要章节将详细介绍 Linux 的详细安装过程。

2.1　版本选择因素

现在 Linux 日益壮大，拥有了越来越多的市场份额，同时也有越来越多的发行版。如何选择适合自己的发行版，让许多刚开始接触 Linux 的用户感到有点无从下手。本节将从包管理系统与桌面系统详细介绍各主要发行版，让读者了解各发行版的特色，以选择最适合自己的系统。

2.1.1　包管理系统

从本质上来说包是 Linux 的应用软件，一般 Linux 系统中所安装的软件均是通过包的方式提供的。包在 Linux 中有着重要的意义。包中除了可执行程序外，一般还包括该包的依赖关系、设置文件等内容。

Linux 软件一个很麻烦的特点就是复杂的依赖关系。所谓依赖关系就是指软件运行必备的条件。如 A 软件运行时系统必需已经安装了 B 类库，如果此时 B 类库没有安装，则称为 A 软件的依赖关系没有满足。如果安装软件时依赖关系没有满足，则软件通常会无法使用。即使有的可以勉强使用，也可能会出现各种奇怪的问题。而卸载时依赖关系如果没有搞清，卸载了系统重要的依赖关系软件，便会导致系统崩溃。而且依赖关系极其复杂，普通用户根本没有能力来理清，所以各大发行版均提供了包管理软件。目前包管理软件市场占有率最高的是 Red Hat 的 RPM 包，而最近发展迅猛的 Ubuntu 则使用 Deb 包管理系统。

1．RPM 包管理系统

Linux 家族中至少有五十余种发行版是基于 RPM 包的，下面仅列出几个比较大的版本，其余版本一般都是从这几个版本衍化而来的。

- SUSE/OpenSUSE：欧洲最流行的发行版，以界面美观而著称；同时具有使用简便，管理方便的特点。著名的视频网站 www.youtube.com 就使用了 SUSE。2004 年，SUSE 公司被 Novell 公司收购，商业版的 SUSE 由 Novell 提供技术支持。而 OpenSUSE 是一个社区发布版，可以自由使用。SUSE 的主页是 www.novell.com/Linux/，OpenSUSE 的主页是 www.opensuse.org。
- Red Hat 企业版：Red Hat 是北美比较流行的 Linux 发行版。这个版本是一个商业版本，免费注册后提供三个月的使用期，三个月后可以付费获得技术支持，主页是 www.redhat.com/rhel。

- 红旗 Linux：我国发布的最有名的发行版，2008 年奥运会在中国召开时，发布了奥运美化版。它以祥云等奥运图案作为装饰，以红色为基调，比较符合中国人的审美观念，主页是 www.redflag-Linux.com。

RPM 全称是 Red Hat Package Manager。所以从名字就可以看出其根源是 Red Hat 公司的包管理方式。后来由于 Red Hat 在业界的领导地位，这种包管理方式被许多大的发行版所采用，成为了 Linux 界的工业标准。RPM 包中包含了软件的可执行程序、相关的配置文件等内容。RPM 包用解压缩程序打开，就可以看到里面的内容，这在提取 RPM 包中的个别内容时比较有效。但是安装 RPM 包时就不会采用这么低效且不安全的方式了，因为 RPM 包有一种高效、安全的包管理程序来进行软件的管理工作。

RPM 管理程序可以提供包的安装、卸载、查询等功能，也提供给开发者打包的功能。在打好的包里有可执行程序及程序的相关安装、依赖信息，这些信息可以方便安装。本书采用的 Ubunu 系统是基于 DEB 包的，所以下面简单介绍几个 RPM 包操作的常用命令：

- 安装 RPM 包：rpm -vih file.rpm。
- 卸载 RPM 包：rpm -e file.rpm。
- 查看 RPM 包的依赖关系：rpm -qpR file.rpm。
- 查询系统已安装的 RPM 包：rpm -q file。

2．基于 Deb 的系统

RPM 的包虽然应用很广泛，但其也有很大的弱点，就是很难解决包依赖的问题。而基于 Debian 的发行版有了更好的选择，就是使用 Deb 包。有消息称 Red Hat 也将逐步支持，以至于使用 Deb 包。一些使用 Deb 包的发行版如下。

- Debian：被认为是最符合 Linux 精神的发行版。它完全是依赖社区的合作精神开发出的极其复杂庞大的 Debian 系统。发行版以稳定而著称，拥有一个巨大的软件库，里面包含有 1 万多个自由软件。同样，从名字就可以看出它创建了 Deb 包，其主页是 www.debian.org。
- Ubuntu：基于 Debian 的一个影响巨大的发行版。截至 2008 年，它在 Linux 个人桌面市场已稳坐头把交椅；其继承和发扬了 Debian 自由与分享的 Linux 精神，一直提供最新的发行版供下载，甚至免费提供安装光盘发放给全世界有需求的用户，已经有 Edubuntu（教育发行版）、Kubuntu（KDE 桌面环境）、Xubuntu（Xfce 桌面）等几个子版本；其主页是 www.ubuntu.com。

Deb 包里同样包含着软件的二进制文件、设置文件等，也可以直接将 Deb 解压查看其里面的内容。但一般来说，这种行为是毫无意义的；因为安装 Deb 包有许多更为安全及方便的方法，如 apt、dpkg、aptitude、synaptic（新立得）、gdebi 软件包安装程序、dselect 等。这些安装方法各有特色，下面是这些安装方式的简介。

- apt（Advanced package tool）：高级包管理工具。提供从源列表（如 CD、远程 FTP 站点）中下载 Deb 包的功能，以及进行包的管理工作。一般来说，如果源列表产生变化后，就应该更新源列表，以获得软件的最新版本信息。用如下命令：sudo apt-get update，apt 命令会自动下载源中软件版本信息更新源列表，然后就可进行安装软件工作了。安装时 apt 会自动选择最新版本进行安装，如果版本一样，则优先选择源列表中靠前的源下载文件，所以在写源列表时要将速度快的源写在前面，如 CD 源或国内的源。

- dpkg：通过数据库来管理系统中软件。数据库中包括了所有系统所需的软件信息，保存在/var/lib/dpkg 目录下。每次安装和卸载软件，都会被记录在数据库中，能确保系统与数据库的记录一致。dpkg 是许多包管理工具的基础。
- aptitude：通过提供一个图形界面来管理软件，功能较为强大。它可以进行升级包，安装包等相关的包管理工作，软件的运行界面如图 2.1 所示。

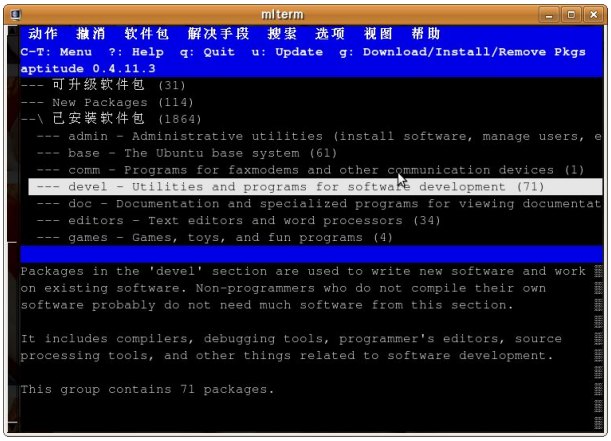


图 2.1　aptitude 界面

- synaptic：（新立得软件包管理器）运行在 X 环境中的包管理软件，在这里可以简单地搜索软件、安装软件、删除软件、升级软件、锁定某个软件的版本，等等。其搜索机制比 apt-cache 要好。并且有多种搜索模式选择。
- gdebi：命令行安装 Deb 的一个命令，相关的一个图形界面安装工具是 gdebi-gtk。
- dselect：类似于 synaptic 的一个管理软件，不过运行在终端，其运行界面如图 2.2 所示。该命令默认并未安装，需要在终端中使用命令“sudo apt-get install dselect”进行安装后方可使用。

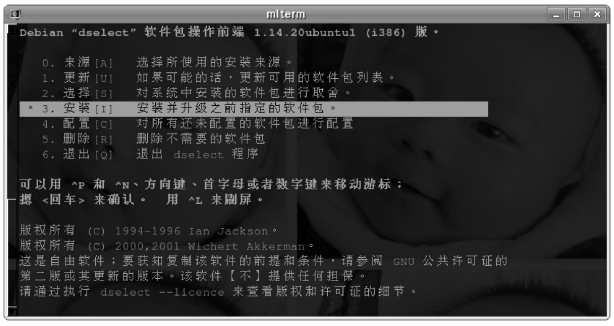


图 2.2　dselect 界面



包是 Linux 操作系统对于软件的统称，现在的发行版对于包管理都有一套成熟的解决方案。由于 RPM 包历史比较长些，所以 RPM 包软件资源可能比 Deb 包软件资源丰富。但随着 Ubuntu 的逐渐流行，越来越多的软件同时提供了 RPM 包与 Deb 包软件供下载使用，所以就算是使用 Ubuntu 也不用担心缺少合适的软件。

2.1.2 桌面系统

桌面系统，是 Linux 呈现给使用者的使用界面。桌面系统直接关系到使用时的难易程度、方便程度，是用户体验的第一道关卡。Linux 的桌面系统主要可分为 KDE 和 GNOME 两种。

1．KDE 桌面系统

KDE 是 K Desktop Environment（K 桌面环境）的缩写。KDE 的特点是界面美观，但相对来说设置和使用稍微复杂。KDE 在 UNIX、许多 Linux 发行版、FreeBSD 上都得到应用，同时，KDE 在非类 UNIX 系统（如 Windows 和 Mac OS X）中也开始发力，试图在计算机桌面界形成统一的标准。

KDE 在 1997 年由 Mathias Ettrich 发布。由于 KDE 的开发采用的是当时并不自由的 Qt 类库，故导致许多开发者担心以后会有版权问题。于是这些开发者另立炉灶，开始了另一个桌面的开发，就是 GNOME。但 2000 年 9 月，Qt 终于发布了 GPL 版的程序库，使那些对 KDE 前途忧心忡忡的开发者再也不用担忧版权的问题了，其主页是 http://www.kde.org。以下的发行版采用了 KDE 桌面作为默认桌面管理系统。

- Debian GNU/Linux：是前面所提到的号称是最忠实于 Linux 文化的一个社区发行版，性能比较稳定。
- Fedora Core：自从 Read Hat 成为商业版后，Fedora 继承了 Red Hat Linux 免费发行的衣钵，专注于个人桌面市场，里面加入了许多的新技术、新功能，但性能不是很稳定。
- Gentoo Linux：能为几乎任何应用程序或需求自动地做出优化和定制。它追求极限的配置、性能，以及顶尖的用户和开发者社区，主页是 http://www.gentoo.org。
- Kubuntu：是 Ubuntu 的 KDE 桌面版，继承了 Ubuntu 的优点，同时具有 KDE 桌面。
- Mandriva Linux：是简单易用的 Linux。
- SUSE Linux：是欧洲流行的 Linux。
- FreeBSD：BSD 是 UNIX 的一个重要分支，比 Linux 更像 UNIX。建议学习大型机的读者先安装一个 BSD 体验一下。FreeBSD 是 BSD 里发展较为迅速的一个版本，主页是 http:/www.freebsd.org。采用 KDE 的 BSD 还有 NetBSD 和 OpenBSD。
- 纯正的 UNIX 也有许多采用了 KDE，如 IBM AIX、Sun Solaris。

2．GNOME 桌面

GNOME（全称 The GNU Network Object Model Environment）可译为 GNU 网络对象模型环境，从名字里可以看出其属于 GNU 的一部分。

一般来说 GNOME 的界面没有 KDE 桌面系统的界面漂亮，这也是其在初级用户中受到冷落的主要原因。但作为一种成熟的桌面系统，它有许多优点，如易用性、稳定性等。易用性是 GNOME 的一个重要特点。另外，它很国际化，从项目发展的开始阶段，GNOME 开发者就极其注重其多语种特性。

GNOME 计划是 1997 年 8 月由 Miguel de Icaza 和 Federico Mena 发起的，采用 GTK 类库。GTK 类库是为设计 GIMP 界面而设计的一个代码库，称为 GIMP Toolkit。由于其使得 Linux 环境的图形界面程序开发更为高效，所以发展很快，成为 GNOME 界面开发的基础。GNOME 最初应用于 Linux 系统，但由于其属于开源软件且性能稳定等优点，所以现在 GNOME 已经运行在大多数类 UNIX 系统上，如 BSD、AIX、IRIX、HP-UX、Solaris 等。但 GNOME 并没有发布

跨平台的版本，如果要在 Windows 平台体验 GNOME，只能将其安装于 Cygwin 中。



其实，桌面系统只不过是 Linux 系统的一个外在表现形式而已，其作用之一是定义了操作软件的方式，像如何打开程序、是单击还是双击等；另一个功能则是定义了程序的界面，是朴素，还是华丽等。它对于软件的使用没有什么影响，Linux 中绝大多数的软件是可以同时运行在任何桌面系统上的。

2.1.3　选择适合自己的发行版

前面介绍了各种 Linux 发行版，各具有不同的包管理方式和不同的桌面系统，可以根据介绍选择自己喜欢的一种发行版使用。但在选择使用哪一个版本时，还要考虑如下的因素。

- 流行程度：选择时应优先选择最流行的发行版。这并不是鼓励用户盲目地追逐潮流，原因是最流行的发行版一般具有最高的人气，在遇到困难时可以得到许多人的帮助。现在个人桌面市场人气最高的就是 Ubuntu。
- 设置复杂程度：虽然 Linux 均是从同一个核心加上一些软件组合而成的，但其中的工作极其复杂。而发行版在使用时，设置的难易程度更是有极大的差别。而如果开始使用时就先进行复杂的设置，无疑会给顺利使用带来很大的障碍。初学者可以选择 Ubuntu。
- 用途：考虑系统的主要用途。主要用作服务器，则要求稳定可靠，最好有技术支持，如 Red Hat 企业版。如果用于家庭娱乐，则要求对多媒体支持程度好、简单易用，如 Ubuntu。
- 界面：喜欢漂亮奢华的外观，选择 KDE，就是 OpenSUSE、Kubuntu。喜欢简洁高效，则 GNOME 是个不错的选择，如 Ubuntu。喜欢自己设计界面，可以下载无图形界面的服务器版，自己安装桌面管理软件，如 Ubuntu 服务器版。
- 机器配置：如果机器配置不够好，选择最新的桌面发行版可能运行不够流畅，就可以选择较低些的版本或者选择专为低配置机器准备的发行版。
- 多语言支持：如果对于英语不怎么精通，那么选择一款界面和帮助均是中文版的发行版无疑是个好主意，如 Ubuntu 或红旗 Linux。

综上所述，一般建议用户选择 Ubuntu，本书就是以 Ubuntu 原型进行介绍的。不过，由于 Linux 之间的兼容性，本书介绍的绝大部分内容也适用于其他 Linux 发行版。



如果参照了这几条，还是不知道该选择哪一个发行版。可以去 http://www.zegeniestudios.net/ldc 做十几道测试题，然后网站会给你一个参考结果，提示应该选择哪一个发行版。

2.2　获得 Linux 安装文件

Linux 是开放源代码的自由软件，所以可以自由获得其安装文件，而不必担心会有任何的版权问题。也就是说无论采用什么方法获得的安装文件均是符合其许可的。而一般的发行版，均在其网站上提供了多种获得其安装文件的方式。如果网络速度较快，可以通过网络下载文件。如果网络速度慢但又希望尽快得到安装文件，则可以购买其 CD。这种购买 CD 的模式和购买 Windows 光盘有着根本的不同，因为购买所需的价格只是 CD 的制作成本费用，而不是 Linux 系统的编写费用。

如果要安装 Ubuntu 系统的话，可以去其网站上申请免费发放的光盘。光盘可以发到中国，只是这种方式让 Canonica 负担的邮寄费用很高。如果自己有刻录机，最好自己刻录，以更好体现共享的精神；把制作和发送 Ubuntu 光盘的钱留给 Canonica，使其开发出更为好用的 Ubuntu 系统。

下面以 Ubuntu 为例，来介绍如何通过上述几种方式获得安装文件。

2.2.1 从官方网站下载光盘镜像文件

Ubuntu 中文主页是 http://www.ubuntu.org.cn/。在该主页左侧可以看到“获取 Ubuntu”的超链接，单击该链接就可进入获取 Ubuntu 的页面。该页面详细介绍了几种获得 Ubuntu 的方式。

“现在下载”提供的是一个 700MB 左右容量的版本，刚好装满一张光盘，里面包括了 OpenOffice.org，GIMP 等常用软件，一般来说可以满足日常工作的需要。如果要获得该版本，简单地单击“现在下载”链接，就来到了下载页面。下载页面提供了几个选项。

1．你想要的版本

在该项中可以提供的选择可能类似这样：

- Ubuntu 8.04 LTS 桌面版本：支持到 2011 年；
- Ubuntu8.04LTS 服务器版本：支持到 2013 年。

桌面版本是预装了 GNOME 桌面系统的版本，普通用户应该选择此版本。而服务器版本针对服务器应用做了优化，但默认是没桌面系统的，需要用户自行下载安装。而支持到 2011 年的意思是在 2011 年以前，系统可以进行自动的升级。

2．您的计算机类型

提供的选择应该类似这样：

- 标准个人计算机；
- 64 位 AMD 和 Intel 计算机。

如果确切地知道自己的计算机是 64 位的 CPU，就可以选择第二项，否则选择第一项“标准个人计算机”。虽然从理论上对于 64 位 CPU 来说 64 位系统性能会好于普通版本，但有的软件并不支持 64 位系统，或者可能有的软件在 64 位机操作系统中使用会出现某些问题，所以推荐普通用户选择标准个人计算机。

3．choose a location near you

意思是选择所在的位置，应该在其中选择离自己最近的服务器进行下载。



服务器一般可选日本、韩国及中国台湾地区的服务器。一般来说这些服务器的速度要快些。

将上面的三项选择好后，就可以单击“Start Download”连接，开始下载了。

但有一点要注意的是：这种方式下载的版本是不包括本地化文件（也就是中文界面内容）的，如中文界面文件、中文输入法等。安装后只有部分界面是中文，其他大部分界面仍是英文的。如果需要下载包含完整中文支持的版本，则应该下载 DVD 版。可能 DVD 的下载量比较小，在 Ubuntu 的网站主页并没有给出 DVD 版本的下载点，但可以在 http://cdimage.ubuntu.com/dvd/current/找到。



在下载 DVD 版本时要特别注意 ISO 镜像文件的名字，一般来说，ISO 文件的名字应该类似这样：intrepid-dvd-i386.iso，该名字对应于标准个人计算机。如果镜像文件名称类似于这样：intrepid-dvd-amd64.iso，则说明该版本是 64 位操作系统版本。可以按照个人的需要进行选择。

DVD 版包含完整的中文支持包及一些常用软件，如 Mysql、Apache 等，还包含了大量的类库。



参照上面的内容可以获得相应的安装镜像文件。获得安装文件后，要使安装 Linux 的过程简单化，应将其刻录在光盘上，然后使用该光盘进行安装。将 ISO 文件刻录在光盘，可供选择的软件有 Nero（Windows 平台）或 k3b（Linux 平台）。

2.2.2　购买 CD 或 DVD

如果网络速度很慢，可能很多用户会考虑购买光盘。Ubuntu 上提供的购买地址对于中国人来说不怎么实用。由于 Ubuntu 是自由软件，可自由分发，故国内有的朋友下载了 Ubuntu 镜像，然后刻录于光盘上，以成本价出售。在淘宝网（http://www.taobao.com）上搜索 Ubuntu，会出现相关的购买信息，可以根据需要选择购买。

2.2.3　索取免费 CD

这是 Ubuntu 的一个特色“不光免费提供一个优秀的操作系统，还刻成光盘免费送上门”。这个做法对于 Ubuntu 的流行是有很大帮助的。打开 https://shipit.ubuntu.com/，单击“Request a CD of Ubuntu Desktop Edition”链接。按照提示，填写各项后就向 Canonical 公司发出了一个一张免费 Ubuntu CD 的申请。一般情况等待一两个月后，就会收到一份国际快件，里面就是 Ubuntu 的光盘。



免费索取 CD 服务是为了方便没有刻录机的用户。而且免费寄送的 CD 包装也很精美，许多人在下载了 CD 后还会去索取一份作为纪念。而且值得庆幸的是，这项服务中国用户也可申请。不过，由于许多人拿到 CD 后会以比较高的价格进行出售，所以现在国内用户申请是比较困难的。

2.3　启动管理器

在正式进入安装进程前，先对启动管理器进行简单的介绍，这也是安装 Linux 重要的预备知识。简单地说，启动管理器是管理系统内所安装操作系统的一款软件。计算机加电启动后最早运行的就是启动管理器。Linux 系统常用的启动管理器主要有 GRUB 和 LILO 两款。

2.3.1　GRUB 简介

GRUB（Grand Unified Boot Loader）是引导管理器。该程序可以从启动时接管计算机，然后根据设置加载各操作系统的内核，启动操作系统。其可管理的操作系统名单很长，基本包括了现在所有的计算机操作系统，如 BSD 系列（FreeBSD、openBSD、NetBSD）、Linux 各发行

版、Windows 系列（Win98、Win2000、WinXP）等。

GRUB 还可识别 FAT32（Win 98 等使用的分区格式）、NTFS（WinXP 常用分区格式）、ext2、ext3、ReiserFS（Linux 使用的分区格式）等分区。也就是说，前面所述 GRUB 支持的操作系统，如果安装在这些分区上均可由 GRUB 直接读取。这就意味着无论怎样它总能找到内核。另外，GRUB 还可以在启动时执行交互操作，可以动态修改 GRUB 启动的内核、启动参数等设置。

在安装系统之前，应该学习相关的启动管理器的知识。这是因为系统安装时会对启动管理器的设置进行更改。如果这种修改造成了不想看到的后果，那么运行相关的设置程序可以将启动管理器的运行拉入正轨。如先安装 Linux，再安装 Windows XP。Windows XP 会修改启动引导信息，导致无法启动 Linux。在这种情况下，就可通过修改启动管理器来实现 Linux 的正常引导。GRUB 对于使用者来说，最重要的有三部分，一是 menu.lst，二是启动菜单的交互式操作，三是 GRUB 的命令行操作。下面分别予以介绍。

2.3.2 menu.lst

menu.lst 是用来显示 GRUB 启动选单的配置文件。在其中可以控制显示的操作系统标题、显示的时间、默认启动的操作系统等内容。在 Linux 系统中，其位于/boot/grub/menu.lst，内容可能类似于如下所示（行号为方便介绍，由笔者所加）：

```
     1 default     0
     2 timeout     10
     3 title       Ubuntu 8.04, kernel 2.5.24-16-generic
     4 root        (hd0,5)
     5 kernel      /vmlinuz-2.5.24-16-generic root=UUID=92ac6170-07a4-4723-
a27d-44864cde4a8a ro quiet splash
     6 initrd      /initrd.img-2.5.24-16-generic
     7 quiet
     8 title       Ubuntu 8.04, kernel 2.5.24-16-generic (recovery mode)
     9 root        (hd0,5)
    10 kernel      /vmlinuz-2.5.24-16-generic root=UUID=92ac6170-07a4-4723-
a27d-44864cde4a8a ro single
    11 initrd      /initrd.img-2.5.24-16-generic
    12 title       Ubuntu 8.04, memtest86+
    13 root        (hd0,5)
    14 kernel      /memtest86+.bin
    15 quiet
    16 title       Other operating systems:
    17 root
    18 # This entry automatically added by the Debian installer for a non-linux OS
    19 # on /dev/sda1
    20 title       Microsoft Windows XP Professional
    21 root        (hd0,0)
    22 savedefault
    23 makeactive
    24 chainloader +1
```

首先来看第 1 行：“default 0”的意思是默认启动第 0 项。在 menu.lst 文件里一个以 title 开头的几行文字称为一项。每一项的编号从 0 开始，依次增加。default 0 说明默认启动的是第 1 项。本例中第 1 个 title 出现在第 3 行，则启动第 3 行开始描述的操作系统。

第 2 行：“timeout 10”，这个时间是指启动菜单停留的时间。这段时间不进行任何操作，系统启动 deault 所指定的项。

第 3 行至第 7 行定义了一个启动项。

第 3 行：“title Ubuntu 8.04,kernel 2.5.24-16-generic”，title 后的字符串定义了该项在启动菜单里显示的文字，这些文字可以自由更改。

第 4 行：“root (hd0,5)”，指示了启动管理器从硬盘哪个分区加载内核，也就是指示“/boot”的位置。hd0 表示第 1 个硬盘，5 则指第 2 个逻辑分区。这里简单介绍一下在 GRUB 中硬盘的表示方法。(hdx,n)是一种方式，hd 表示 IDE 硬盘，hd1 表示第 2 个硬盘，依此类推。“,”后面跟的数字表示分区，一块硬盘允许有 4 个主分区，所以把 0 到 3 留给了主分区。逻辑分区由 4 开始，表示第 1 个逻辑分区，而 5 表示第 2 个分区。

第 5 行：“kernel /vmlinuz-2.5.24-16-generic root=UUID= 92ac6170-07a4-4723-a27d-44864cde4a8a ro quiet splash”，kernel 指示内核文件存放的具体位置。本例中，内核文件名是“vmlinuz-2.5.24-16-generic”，而且内核文件就存放在一个独立分区的根目录中。由于内核文件只能存放在 boot 目录中，所以“/boot”是一个独立的分区。如果 boot 目录安装在“ / ”根分区，则内核的位置可能像这样：“/boot/vmlinuz-2.5.24-16-generic”。这行的 root 是指 Linux 根目录“/”所在的分区位置，该分区是用分区的 UUID 号码来指定的。ro 意思是只读。

第 6 行：“initrd /initrd.img-2.5.24-16-generic”，意思是加载 Linux 格式的初始化虚拟盘，并设置必要的参数。

第 22 行：savedefault 将当前选项存为默认值。

第 23 行：makeactive 将 root 设备置为活动分区。当然，此命令只对 PC 的硬盘主分区有效。

第 24 行：“chainloader +1”指示 GRUB 不要试着自己加载 Windows 的引导程序，而应该将程序执行位置调整到 C 盘的首扇区，该位置包含 DOS/Windows 启动程序。

其他行的理解可参照这几行的介绍。知道了这个文件各项的含义后，就可以按照自己的喜好修改了。如本例中如要默认启动 Windows 系统，就可以将第 1 行改为 default 4。

2.3.3　启动菜单的交互式操作

计算机打开电源后加电、自检，然后就运行到 GRUB 启动菜单上停下来，如图 2.3 所示。

现在可以用方向键上下移动选项，以启动不同的操作系统。也可以进行一些交互操作，如按 E 键编辑启动设置，按 C 键进入命令行模式。

1．编辑启动设置

如果启动时觉得某些项（每项代表一个操作系统）的设置有问题需要修改，就可以将选项条移动到该项上，然后按 E 键，就进入了对该操作系统启动设置的修改模式，如图 2.4 所示。

由图 2.4 可以看出，可以修改的项有三条。第一条是主目录的 UUID 号，第二条是核心的位置及名字，第三条是初始化的镜像文件。该模式下可进行如下操作。

- b：启动本项。
- e：编辑当前选择条目。

- c：进入命令行模式。
- d：删除所选行。
- o 或 O：添加一个新行用于输入某些命令。

编辑完成后，可按 B 键（boot）启动该操作系统，就可以使用刚刚修改的内容启动系统。如按下 Esc 键返回，则不会对启动项做任何改变。

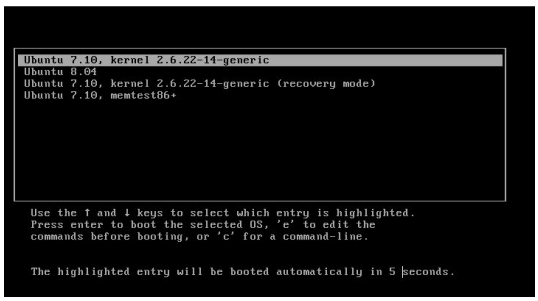


图 2.3　GRUB 启动菜单



图 2.4　修改特定项的启动设置

2．命令行模式

在图 2.3 所示与图 2.4 所示界面中按 C 键均会进入命令行模式。该模式允许用户输入 GRUB 命令来执行特定的操作。命令行模式的操作如图 2.5 所示。

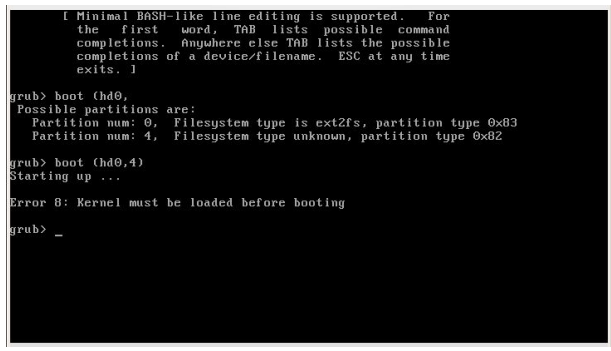


图 2.5　GRUB 的命令行模式

GRUB 的命令行模式的提示符是“grub>”，可以在其后输入 GRUB 命令。GRUB 可用的命令将在 2.3.4 节中进行介绍。

2.3.4　命令行操作

其实 menu.lst 中的文本，除了 title 外，其余均是 GRUB 的命令。这些命令均可以在 GRUB 的命令行中使用。要使用 GRUB 的命令行，一种方式是在系统启动时进入命令行模式，参见 2.3.3 节相关内容的介绍；另一种方式是在终端里输入 GRUB 来运行 GRUB 的命令行。在命令行中可以运行如下命令。

- boot：启动操作系统，或者启动上一次的 chainloader。
- chainloader：启动特定的文件或者一个 chainloader。如果启动的文件位于特定分区的第一个扇区，可以用“+1”符号代替文件名称，如“chainloader+1”表示从 0 开始读一个

扇区。

- displaymem：显示系统内存容量。
- install：安装 GRUB 程序本身。如在安装前将 GRUB 的 stage1 和 stage2 文件复制到“/boot/GRUB”中，然后运行这个命令就会自动安装 GRUB。如要将 GRUB 安装到第一块硬盘的第一个主分区中，可用该代码“install(hd0,0)”。如果没有进行 stage1 和 stage2 的复制工作，在安装时就需要指定 stage1 和 stage2 的位置。
- kernel：指定内核文件，后面跟内核参数。
- root：指定根分区，并挂载该分区。
- rootnoverify：指定根分区，但是不挂载该分区。
- foreground：设置图形模式下的前景色。
- halt：关机。
- quit：退出 GRUB。
- password：给 GRUB 设置密码，未得到授权的人就不能随便修改 GRUB 配置了。

2.3.5　应用实例

GRUB 在实际应用中，有许多非常实用的技巧。如可以用 GRUB 引导 Windows 系统，也可以用其来修复硬盘的分区表。下面以两个例子来说明其功能。

1．引导 Windows 系统

可以在 GRUB 命令行输入如下代码引导 Windows（假设 Windows 安装在第 1 个硬盘的第 1 个主分区，在 GRUB 中代号为“hd0,0”。一般先安装 Windows，在安装 Linux 的系统中 Windows 均安装在 hd0，0）：

```
grub>rootnoverify (hd0,0)
grub>chainloader +1
grub>boot
```

2．修复硬盘的分区表

有时候，在安装双系统的电脑中，由于 Windows 出了问题，会重新安装 Windows。但这样会破坏原来由 GRUB 管理的 MBR（磁盘分区表），Linux 无法通过菜单启动。但这种情况下 Linux 系统并没有受到损害，只是没有 GRUB 的引导，无法启动。只要用 Ubuntu 的 Live CD 启动系统，然后进入命令行模式，使用如下命令就可以修复 MBR。

```
grub>root (hdX,Y)
grub>setup (hd0)
```

2.3.6　另一款启动管理器 LILO

LILO 是早年 Linux 默认安装的启动管理器。由于其在灵活性和性能方面均要逊色于 GRUB，所以现在的发行版都不再预装 LILO，只在有些嵌入式 Linux 中有些应用。LILO 与 GRUB 一样，都是通过一个配置文件来完成配置的。不过 LILO 的配置文件位于/etc/lilo.conf，里面各项的含义也与 GRUB 有相似的地方。LILO 的参数项与参数值之间要有等号，而 GRUB 中是不用等号的。下面是一些 LILO 常用的命令及其功能介绍。

- boot：指定系统内核所在的分区。
- map：指向引导期间 LILO 内部使用的映射文件，其中包含有描述符表。
- install：指示 LILO 自身的位置。
- prompt：使用菜单提示用户选择。
- timeout：指定使用默认操作系统之前的等待时间，单位是十分之一秒。
- compact：压缩选项，将连续的读磁盘的请求合并为一个单独的请求。
- default：指定默认启动的操作系统。
- root：指定系统文件所在的分区，如何读懂分区信息，请参阅本章相关内容。
- password：指定启动某个操作系统的密码，以明文方式保存。

2.4 丰富的 Linux 安装方式

Linux 各发行版为了让用户可以更容易体验 Linux 的性能，提供了许多将其安装进计算机的形式。要将其安装至还没有任何操作系统的计算机里是没有问题的；将其安装到已经安装了 Windows 的系统里，也没问题；甚至可以将其作为 Windows 的一个程序，安装在 Windows 中，可以由 Windows 轻易卸载。本节就简单地介绍一下使用各种方法进行安装前的准备工作。由于完成了准备工作后，后续的正式安装过程均是一致的，所以具体的安装过程，将单独在本章 2.6 节中进行详细的介绍。

2.4.1 使用 Wubi 安装

使用 Wubi 进行安装是使用一个叫 Wubi 的 Windows 程序来实现的。这个程序的名字听起来和五笔有点像，但其实他们之间没有什么关系，而是得名于“Windows-based Ubuntu Installer”（翻译过来就是基于 Windows 的 Ubuntu 安装器）。Wubi 是使用 GPL 许可的自由软件，其开始是独立项目，后来得到 Ubuntu 的官方支持，在 Ubuntu8.04 版的 Live 光盘里就附带了这个程序。但有点奇怪的是 8.04 的 DVD 版中并未提供该软件，需自行下载，下载的地址是 http://releases.ubuntu.com/8.04/Wubi.exe。

使用 Wubi 进行安装是将 Linux 作为 Windows 系统的一个程序进行安装的。这种方法的思路是将 Linux 像一个普通应用软件一样安装到 Windows 的某一个分区中，启动时选择进入 Linux 就可以运行 Linux 系统了。而安装了 Linux 的分区甚至还允许存放其他文件。这种方法不改写任何硬盘信息、不用格式化硬盘、不用重新对硬盘分区、安装简易、卸载容易，是新手上路时一个比较好的选择。

Wubi 虽然简单，但其性能没有采用直接安装方式的性能高。而且由于其文件系统并没有独立放在一块分区中，而是放在 Windows 文件系统中，所以文件系统更容易发生问题。

具体的安装过程比较简单，和安装其他 Windows 软件没有什么两样，只要简单地按照下文所述步骤进行就可以了。

（1）双击下载的 Wubi.exe 文件，出现如图 2.6 所示安装设置对话框。

（2）在“安装驱动器（4GB 可用）:”下拉框中选择要安装 Ubuntu 的分区，如选择“C:”等。注意要选一个剩余空间较大的分区，Ubuntu 安装至少需要 3GB 左右的空间。

（3）在“语言”下拉框中选择“Chinese（Simplif…）”，将界面语言设置为中文。

（4）“桌面环境”下拉框仍然保持默认的 Ubuntu 是个不错的选择。

（5）在“用户名”文本框中填写登录 Ubuntu 时用的名字。默认是管理员 administrator。



图 2.6　Wubi 安装界面

（6）在“密码”下面的两个文本框中输入两次同样的密码。

（7）设置完成后，单击“安装”按钮进行安装。但这时其实没有开始真正的安装，而只是在 Windows 中创建 Ubuntu 的虚拟硬盘文件。几分钟后，虚拟硬盘文件创建结束。

（8）创建虚拟硬盘结束后，Wubi 会提示“正在完成安装向导”，并要求重启计算机。单击“是，现在重新启动”按钮，然后从光驱中取出光盘，开始重启。

（9）计算机重新启动后，会发现启动选单中多了一个名为 Ubuntu 的选项，选择该选项才会正式进入 Ubuntu 的实际安装。实际的过程与本章 2.6 节相关内容是一致的，如果正在使用 Wubi 安装，请跳转至 2.6 节查看如何进行其余的安装过程。

这样安装的 Ubuntu 更像是 Windows 的一部分。如果不想使用，还可以像卸载普通软件那样在 Windows 中将其删除。删除的方法也很简单，运行 Wubi 目录里的 Uninstall.exe，就可以轻易地删除 Ubuntu 了。

2.4.2　使用光盘启动安装

这种安装方式是最容易安装的方式之一。安装时只需要在打开电脑电源之前将 Linux 光盘放入光驱，然后选择从光盘启动就可以进入 Linux 安装界面了。Ubuntu 8.04 版 DVD 的启动画面如图 2.7 所示。

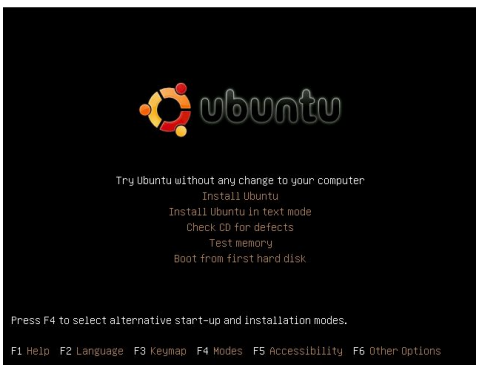


图 2.7　Ubuntu 光盘启动画面

如果选择第一项“Try Ubuntu without any change to your computer”，就可以不用安装直接运行 Ubuntu。本项就是所谓的 Live CD 版。这种方式不在硬盘上写入任何信息，对 Linux 操作系统所做的设置也不会被记录，对于想要试用 Ubuntu 的人来说，是个不错的选择。采用这种方式不但可以体验 Ubuntu 的性能，而且在启动成功后，在 Ubuntu 桌面上还有一个安装 Linux 的快捷方式，双击该快捷方式即可进入图形界面的安装。

第二项“Install Ubuntu”：安装 Ubuntu，选择该选项后即可进入安装过程，详情可查看 2.6 节。

第三项“Install Ubuntu in text mode”：在文本模式下安装，这个安装方式比第二项复杂一些，不建议新手使用。

第四项“Check CD for defects”：检查安装 CD 的质量。

第五项“Test memeory”：检查系统内存是否有错误。

第六项“Boot from first harddisk”：如果不想从这个光盘安装新的 Ubuntu，而想用原来硬盘中安装的操作系统，就可选择这个选项。



一般安装时可以选择第 2 项，该项由于不用启动 Ubuntu live 版，所以速度会稍微快一些。第 3 项文本模式相对来说速度是最快的，但对初学者来说操作（尤其是硬盘的分区）是比较困难的，所以不推荐普通用户使用。

2.4.3 从硬盘安装

当得到一个 Linux 的光盘镜像文件（ISO）后，将其刻录入光盘，就可以采用光盘启动的方式进行安装。但是，有的用户并没有刻录机，难道因为这个原因要再购买一台刻录机吗？当然不用！只要得到了 ISO 文件，再加上一个可以启动的操作系统（如 Windows XP 系统）就可以进行从硬盘安装。

其实现原理比较简单：一般的操作系统均有启动管理的功能，就是说系统中如果存在多个操作系统，启动管理器会自动显示这些操作系统，供用户选择适当的操作系统以便启动。这种方式对于 Windows 系列（如 Windows 98、Windows XP 等）都是有效的。但 Linux 由于其内核与 Windows 不一致，所以 Winodws 无法正确识别 Linux 的启动代码。

如果启动管理器能识别 Linux 的内核文件，那就简单多了，直接将 Linux 内核文件 initrd.gz 和 vmlinuz 放在硬盘中某个目录，然后在启动管理器中指定该位置，就可以用这两个内核文件启动 Linux 安装过程了。但遗憾的是 Windows 并不识别 Linux 的内核。这样工作就稍微复杂了一些，需要先用 Windows 的启动管理器启动另一个启动管理器（如 GRUB），这个管理器可以识别 Linux 启动程序，然后用 GRUB 来启动 Linux 内核。



从硬盘安装比较复杂，不太适合没有实践经验的普通用户，而且经验证明，最好下载 Alternate 版本的光盘镜像文件，其他版本的文件有时会无法进行安装。

安装前的准备工作，包括下载 Ubuntu 的光盘镜像文件（可参照 2.2.1 节内容），以及下载 GRUB020p.zip（这是启动管理器，也可使用 grub4dos），还需要一个 FAT 格式的磁盘分区（用来安装 Linux，本例以 Windows 的 E 分区为例）。下面介绍详细的安装过程。

（1）用虚拟光驱加载 Ubuntu 的光盘镜像文件。

（2）在“我的电脑”中打开这个虚拟光驱中的 install 目录（有的光盘是 isoLinux 目录），将内核文件 initrd.gz 和 vmlinuz 复制到硬盘分区 E 的根目录。

（3）将 Ubuntu 的光盘镜像文件（文件名类似于***.iso）复制到 E 盘的根目录。



E 盘在 GRUB 中该盘表示为（hd0，5）或（sd0，5）。

（4）现在 E 盘就应该有三个文件，一个是 Ubuntu 的光盘镜像文件，一个是 initrd.gz，另一个是 vmlinuz。

（5）下载 GRUB020p.zip，解压到“C:\”下，得到一个 grldr 目录与 boot 目录。这两个目录包含了引导系统所需的文件。

（6）编辑 Windows XP 系统启动管理文件“C:\boot.ini”（由于该文件具有只读属性，无法立即修改，在编辑前需要取消该文件的只读属性）。在 boot.ini 文件内容的最后加入一行“C:\grldr="Ubuntu by GRUB"”，加入后 boot.int 应该看起来像这样：

```
[boot loader]
timeout=10
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Professional" /noexecute=optin /fastdetect
C:\grldr="Ubuntu by grub"
```



加入这一行的目的是在 Windows 启动管理器列表中添加另一个可以识别 Linux 内核的启动管理器，其目录位于 C:\grldr，在菜单中显示的内容为 Ubuntu by GRUB。

（7）将 timeout=0，改为 timeout=10，这是让菜单显示的时间由 0 秒延长到 10 秒，用户就可以选择进入 Linux 安装项。

（8）再编辑“C:\boot\GRUB\menu.lst”，在其中加入：

```
title Ubuntu Install
kernel (hd0,5)/vmlinuz root=/dev/ram ramdisk_size=32000 devfs=mount,dall vga=791
initrd (hd0,5)/initrd.gz
boot
```

这个文件是 GRUB 启动管理器的菜单，GRUB 可以加载 Linux 内核，在本例中是由 Windows 启动管理器 ntldr 启动的，添加这几行的目的是指定 Linux 内核所在的位置。vga=791 指定显示模式，如果出错，可以用 GRUB 交互的方式将其改为 792（vga=792）或 771（vga=771）。

（9）重新启动计算机，会出现一个选单，用来选择要启动的操作系统。应该选择“Ubuntu by GRUB”选项（这是第 6 步设置的操作系统的名称），意思是进入 GRUB 启动管理器。

（10）同样 GRUB 也提供了一个选单，选择“Ubuntu Install”选项，GRUB 就会寻找 Linux 内核文件，然后加载，就开始了正常的安装过程，接下来的步骤可以参见 2.6 节内容。



进入安装界面后，其他设置均很简单，但有一点要注意，就是分区必须选择手动分区。最少需要 2 个分区：SWAP，虚拟内存分区和“/”分区。分区大小可以参照 2.5.3 节内容。

2.5 Linux 分区与挂载点

要安装操作系统，一般来说要对磁盘进行分区等操作。磁盘也称硬盘，由于其容量大，价格便宜，成为计算机操作系统中最重要的存储设备。而分区呢，就是指将硬盘分为一些块的工作。Linux 对待硬盘的分区有其独特之处，将分区称为挂载点。本节将介绍这些内容。

2.5.1 硬盘的分区简介

因为硬盘容量很大，现在动辄以几百 GB 来计其容量，如果将所有文件都放在一个分区中，有两个问题，一是重要的内容和不重要的内容放在一起，发生问题后，重要的数据也会遭到破坏；二是如果文件太多，会很难寻找到需要的资料。如果将其进行分区，这些问题就迎刃而解了。可以将一块硬盘分为许多分区，一个分区用来存放重要的系统文件，一个分区用来存放机密的文件数据，另一个则可以存放普通用户的数据。

硬盘的分区分为主分区和逻辑分区，一块硬盘的主分区最多只能有 4 个，这些分区是用来存放操作系统引导数据的。但并不是说一个硬盘最多只能安装 4 个操作系统，因为只有 Windows、FreeBSD 等才需要将自己安装在主分区中，而 Linux 可以安装在逻辑分区中。

2.5.2 Linux 分区编号的意义

在 Linux 中，分区是这样表示的：/dev/hda、/dev/hda1、/dev/sdb1 等。这里，以/dev/hda1 为例来说明其含义。前面提到，Linux 将所有存储设备均作为一个子目录来管理，/dev 就是 Linux 中所有存储设备存在的目录，hd 表示这是一块 IDE 硬盘（harddisk）。如果是 SATA 硬盘、USB 设备则用 sd 来表示。第三位的字母 a 表示这是第一块 IDE 硬盘，依此类推，hdb 表示第二块 IDE 硬盘等。数字 1 ，表示这是该硬盘中的第 1 个主分区，主分区有 4 个，依次可以是 hda1，hda2，hda3，hda4，而第一个逻辑分区则用 hda5 来表示。

因为在 Linux 中，为了避免分区编号的交错，分区的顺序是不能改变的，分区标识则由它们在硬盘中的位置决定。系统又要为所有可能的主分区预留标识一般系统中只有一个主分区，所以经常会看到系统的分区是 hda1，然后直接就是 hda5。所以 1～4 一定不会是逻辑分区，5 则是第一个逻辑分区，依此类推。

2.5.3 Linux 系统的挂载点

在 Ubuntu 系统的安装过程中，需要选择系统目录的挂载点。所谓挂载点，就是将分区挂载到一个以“/”开头的目录中。如新建了一个分区，设置其挂载点为“/boot”，则访问“/boot”目录时，实际上是在访问新建的分区。从理论上说每一个以“/”开头的目录均可成为挂载点。

安装 Windows 时，是把系统安装在某一个分区，从某种意义上来说，是把系统挂载到分区上。而在 Ubuntu/Linux 中则相反，是把分区挂载到系统中。在安装过程中，系统安装程序会建议一些挂载点，这些目录挂载点有其特定的用途。下面来介绍这些挂载点的功能与应该给其分配的空间大小。

- /：根目录：必须挂载的目录，是所有其他目录的基础。Ubuntu 安装后大约占 2～3GB 的空间，如果再安装一个桌面环境等，应该占用 5～6GB 的空间。如果要经常使用 Ubuntu，应该给其分配 10GB 左右的空间。

- swap：交换分区，相当于 Winodws 中的虚拟内存，不过在 Linux 中将其分离出来，直接放在独立的分区上。一般来说，内存小于 1GB 的，交换分区可分配内存容量的 2 倍左右。而内存容量大于 1GB，交换分区分配 1GB 也就足够了。
- /home：是用户的主目录，用于存放自己的文件。按普通桌面用户的使用经验来说，应该给其分配一个独立的分区，而且是最大的一个分区。这样如果要重装系统，就不至于破坏自己的资料文件，而且由于许多软件的配置文件均放在这个目录中，重装后就不用再花费精力重新配置了。
- /usr：应用程序目录，大部分的软件都安装在这里。普通用户可不再划分这个分区。
- /var：安装的是服务器方面的应用，同样，普通用户可不划分这个分区。
- /boot：存放的是系统启动所需的内核文件，有时也包括启动管理器。建议普通用户也划分本分区，因为如果系统文件损坏，若/boot 中的内核文件完好的话，可以将计算机启动然后检查问题出在什么地方。一般来说，Linux 内核只有不到 20MB，所以给该分区分配 100MB 大小就足够了。一般来说，分配 2 的整数幂大小，如 128MB。

2.6　安装过程详解

这里主要是介绍图形界面的安装，因为对初学者来说，图形界面是最容易理解和适应的方式。而如果理解了安装过程，文本界面与图形界面安装本质上是一致的，在熟练后使用文本界面安装程序也是件很容易的事。图形界面安装一共只需 7 步。

2.6.1　分区前的基本设置

安装一共只需要 7 步，其中前三步只是进行一些基本的设置，比较简单。本节将这些步骤按逻辑意义分为三个部分进行介绍。



由于 Linux 安装过程会对硬盘进行一些操作，所以在 Linux 安装前，最好先将硬盘中重要的资料进行备份。万一执行了错误的操作，导致硬盘数据丢失，也不至于有太大的损失。

1．选择安装语言

这是安装 Linux 的第一步，界面如图 2.8 所示。在左侧的列表框中，向下拉动滑动条，直到出现“简体中文”选项，然后选择该项，意思是安装界面和安装后系统默认的语言为中文。单击“Forwoard”（前进）按钮，到下一步。

2．选择时区

如果显示不是 Shanghai（上海），需要在“所选城市”下拉列表中选出上海或者直接在世界地图中选择上海的位置，直到所选城市变为上海，时区成为 GMT+8。继续单击“前进”按钮到下一步。

3．键盘布局

这个很简单，就是指键盘上的键对应于那一个字母，对于英语国家，是没有什么区别的，但有些其他语种可能需要特别的键盘，如法语键盘、日语键盘等。中国用户一般直接接受系统默认值（左边列表中选中 USA，右边列表中选中 USA）即可。然后单击“前进”按钮到下一步。

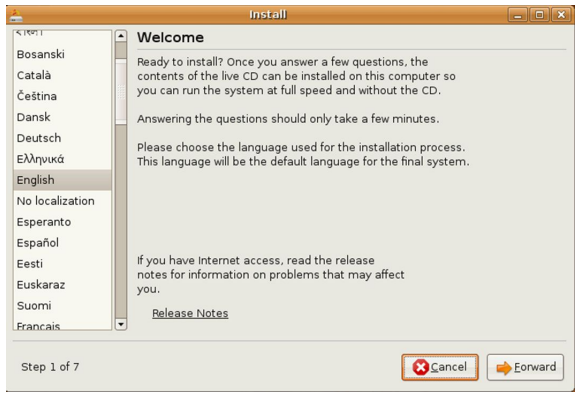


图 2.8　语言选择对话框

2.6.2　硬盘分区

这是极其重要同时也容易出现问题的一个环节，合理的分区会提高系统性能以及安全度。而且如果分区过程中出现错误，也容易导致原有数据丢失，所以这一步需要仔细认真地查看说明。现在出现的对话框应该为如图 2.9 所示。

1．预备硬盘分区

第一项“Guided - resize SCSI 1(0,0,0)，第 3 分区(sda) and use freed space”的意思是主硬盘分区已经安装有一个操作系统，Ubuntu 使用该分区剩余的空间安装。

如果磁盘有多个分区，本项可能是“向导-使用最大的连续空闲空间”，然后下面列出可供使用的分区列表。如果计算机里已经安装有一个操作系统，比如说 Windows XP，而且在安装 Linux 前，已经给 Ubuntu 保留了一个分区，就可以使用该项，选择那个保留的分区进行安装。

第二项“向导→使用整个硬盘”的意思是使用整个硬盘进行安装。如果只有一块硬盘，则选择此项会将硬盘中原有数据清空。有多个硬盘，选择此项会将所选硬盘中原有数据清空。

前两项都是自动分区，如果选择前两项中任一项，则系统会自动根据需要划分需要的分区，比较方便，但不够灵活。



图 2.9　硬盘分区

第三项“手动”是手工完成分区工作，包括分区的数量、大小等。这样，可以对分区有更

精确的控制，也会让系统获得更好的性能，所以推荐使用本项进行分区。

2．手动分区

选择第三项“手动”，单击“前进”按钮，进入实际的分区过程。手动分区的主界面如图 2.10 所示。表中列出了系统中所有的硬盘及该硬盘的分区情况。还显示了各分区的大小、分区类型、已经使用的空间等信息。可以根据这些信息来决定将 Linux 安装到哪一个分区。



图 2.10　准备分区

3．删除分区

删除分区是指将硬盘中已经划分的分区删除，使之成为未分配状态的过程。分区删除后，其原始内容会全部消失。下面以图 2.10 为例介绍将硬盘的原有分区（sdb6）删除的具体过程。

（1）单击要删除的分区，使其处于选中状态。本例选中 sdb6。

（2）单击“Delete partition”按钮，将第 1 步选中的分区 sdb6 删除。

4．新建分区

要使硬盘可以存储文件，需要先建立分区。要建立分区，硬盘必须要有尚未分配的空间。在 Linux 中，新建分区需要设置分区的格式、分区的大小等参数。要在还没有建立分区的硬盘空白区域建立分区的方法也很简单，只需如下三步。

（1）选中要建立分区的硬盘空白区域。

（2）单击“New partition”按钮，即可开始创建新分区，弹出如图 2.11 所示的对话框。



图 2.11　创建分区

该对话框中有许多设置内容，用来设置分区的各种属性，如分区的大小、分区的格式等。下面将各项的意义列于下，以便理解。

- 新分区的类型：用来选择分区的类型，只有挂载点“/boot”选择“主分区”单选按钮，其他挂载点均选择“逻辑分区”单选按钮。
- 新建分区容量：输入新分区的大小，单位是兆字节，可参照 2.5.2 节内容给各分区选择适当的容量。
- 新分区的位置：一般选择“开始”单选按钮。
- 用于：（这个翻译不太恰当，应该是“使用”），其实是指分区的格式，除了 swap 挂载点

使用 swap 选项外，其他挂载点一般选择“Ext3 日志文件系统”选项。

- 挂载点：从某种意义上说，其概念相当于 Windows 中的分区，参照 2.5.3 节内容输入即可。

（3）设置结束后，单击“OK”按钮完成新建分区的操作。

参照以上方法，依次建立 Linux 安装所需的各分区。分区至少要有两个，分别设置为 swap 和“/”挂载点。推荐建立 4 个分区，除了 swap 与“/”，再分别挂载一个“/boot”，以及一个“/home”。笔者的分区如下所示：

```
Device Type Mount point Size
/dev/sdb1 ext3 /boot 131MB
/dev/sdb5 swap 386MB
/dev/sdb6 ext3 / 10996MB
/dev/sdc5 ext3 /home 11424MB
```

这样分区的理由 2.5.3 节已经详细阐述过，这里不再赘述。建立分区工作完成后，单击“前进”按钮到下一步。

2.6.3 分区后的确认步骤

现在计算机基本上已经做好准备安装 Linux 了。不过，在正式开始安装前，还需要进行一些简单的确认步骤，包括建立个人信息，以及安装前的确认等内容。

1．建立个人信息

对话框上给出的说明很清楚，可以参照说明进行填写。需要注意的是，“你想用什么名字作为登录名？”下的文本框输入的名字才是登录系统的用户名。然后选择一个复杂并且容易记忆的密码输入两次，其他的文本框可随意填写。然后单击“前进”按钮。

2．迁移文档及设置

如果系统检测出需要导入的文档之类的东西，会提示导入，只需按照提示操作就行了。然后单击“前进”按钮。

3．安装前的确认工序

前面所修改的硬盘分区信息其实还未真正应用于硬盘。现在 Ubuntu 给出了前几个步骤设置的各项及其要对硬盘进行的操作，如图 2.12 所示。如果发现有什么问题可单击“后退”按钮来更改。要出现启动选单，就应当将 GRUB 安装到硬盘来接管启动过程。方法如下。

（1）单击“高级”按钮，弹出“高级选项”对话框。

（2）在出现的“高级选项”对话框中，确保选中其中的“安装启动加载器”选项。

（3）单击“确定”按钮完成。

检查一下各种设置，如果没问题，就可单击“安装”按钮开始正式的安装工作。



如果在安装中出现“正在扫描镜像站点”字样，然后停下来不动，那是系统正在 Ubuntu 的镜像服务器上下载中文支持文件。由于该文件比较大，而系统默认的服务器一般速度较慢，所以这个过程极其耗费时间。这时请拔掉网线，强行终止该过程，系统会继续其他工作。别担心，在安装完之后，自己配置一个比较快的源，然后更新中文支持会快得多的。

这个安装过程视电脑配置不同而略有不同，现在主流配置机型大约 10 多分钟就可以安装完成。

而且在剩下的安装过程中，不需要再进行任何设置了。放松一下，喝杯咖啡，听听音乐。



图 2.12　硬盘在实际分区前的确认

等一会儿，系统会显示一个对话框，其内容是“安装完成，是否重启”。单击对话框的“重启”按钮，安装系统会将光驱弹出，把 Linux 系统光盘从光驱中取出，然后按下 Enter 键，系统会进行重启。系统重新启动后，就可以进入到 Linux 世界中了。

2.7　小结

本章介绍了 Linux 各种版本的特点，以及安装 Linux 的各种方法。本章的重点内容是 2.6 节所介绍的详细的 Linux 安装过程，掌握了该部分的内容，就可以熟练进行各种 Linux 发行版的安装了。同时 Linux 分区与挂载点的内容是安装的基础，应该先行掌握。本章最难掌握的内容是 2.3.4 节所介绍的 GRUB 相关内容，但 GRUB 是 Linux 启动的基础，而且其具有很强的实用性，应该加以掌握。

第 3 章　桌面环境简介

现在，Linux 发行版中风头最劲的 Ubuntu 已经成功地在计算机里安家。但对于 Windows 的用户而言，对于 Linux 的操作可能还不太适应。一般来说，用户使用 Linux 会从桌面环境开始，故本章是入门读者的指引章节，主要介绍 Linux 系统重要的桌面环境 GNOME。阅读本章后，读者应该能顺利地将 Windows 的相关操作经验平滑过渡到 Linux 系统 GNOME 的操作中。

3.1　系统登录

要想领略 Linux 的强大功能，首先要登堂入室。至于如何“入室”，方法很简单，只要过两关。第一关叫做启动选单，是在启动选单中选择正确的启动项目，这样就可以来到 Linux 的大门前；第二关叫做登录，通过 Linux 验证，就可以推开大门，正式进入 Linux 的世界了。首先来看第一关。

3.1.1　启动

按下电源按钮，屏幕上闪过熟悉的自检信息，然后会显示类似于如图 3.1 所示的画面。这就是 GRUB 启动管理器发挥作用了，屏幕上显示的是计算机中安装的所有操作系统。这个选单会随着计算机中安装系统的不同而有差异。

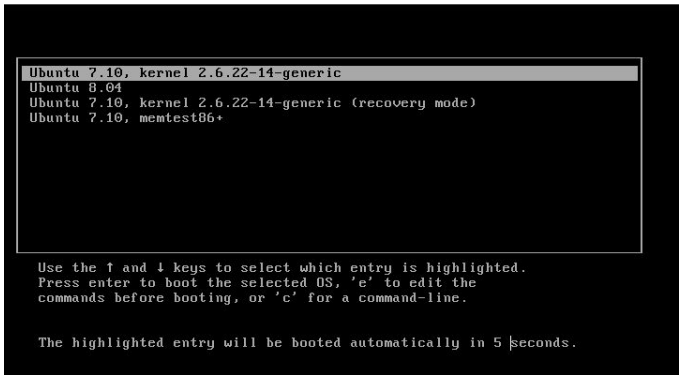


图 3.1　GRUB 启动选单

方框中白色底子的第一行表明当前选项为 Ubuntu，内核版本是 2.6.22-14；第二行表明系统中还装有 Ubuntu8.04；第三行是系统出现问题后修复时使用的选项；第四行是运行内存测试。使用键盘上的上下方向键，表示选中的白色状态条会上下移动。移动该状态条到要启动的选项，然后按下 Enter 键就会启动该选项所代表的操作系统。本次操作选择第二项。方法是按下方向键，使白色长条停留在“Ubuntu 8.04”行，再按下 Enter 键，就会进入 Ubuntu 8.04。

这时，系统会加载系统启动所需要的软件，如操作系统内核、硬件驱动、各种服务等。加

载的细节对于普通用户来说是没有必要知道的，因此，Ubuntu 系统在屏幕上显示了一张 Ubuntu 标志的图片，而并不是显示加载的细节。如果想知道细节，可以按下 Ctrl+Alt+F1 组合键查看。当进程加载成功一项服务或程序时，就会在屏幕上显示一行该加载项的简单说明。如果加载成功，会在该行的最后出现“[OK]”的字样。系统需要的软件加载完成后，就会进入登录窗口。

3.1.2　登录

Ubuntu 的初始登录窗口很简洁，如图 3.2 所示。中心一个大大的 Ubuntu 标志加上 ubuntu 字样，中心稍偏下有一个文本框，其上是一个提示文本框输入内容的标签。在窗口的左下方，有一个 Options（选项）按钮，单击该按钮会出现一个菜单（即图 3.2 左下角显示的菜单）。菜单项内容由上至下依次为 Select Language（选择语言）、Select Session（选择会话）、Remote Login via XDMCP（用 XDMCP 远程登录）、Restart（重启）、Shut Down（关机）、Suspend（挂起系统）。



图 3.2　登录窗口

1．更改系统语言

选择菜单中的“Select Language”命令，会弹出一个对话框，在标签“Select the language for your session to use”（选择会话中使用的语言）下面有一个下拉列表框，单击该列表框的下三角按钮打开下拉列表，选择其中的“简体中文”选项，然后单击 Change Language（改变语言）按钮确认更改，菜单语言会变为中文显示。

2．更改当前会话

选择菜单中的“Select Session”命令，会弹出一个对话框，其功能是改变会话。所谓会话，就是登录后进行什么工作。如在其中选择“2.GNOME”单选按钮，则意味着选择了 GNOME 会话，单击“Change Session”（更改会话）按钮，会在登录后进入 GNOME 桌面系统。

如果选择“Failsafe Terminal”单选按钮（安全模式控制台）会在登录后进入到安全模式的控制台，可执行相关的控制台命令进行系统修复的操作。

如果安装了其他桌面系统如 KDE 或 FVWM，则这些桌面系统也会作为一项会话在这个 Sessions 列表中显示出来。如需要更改（一般不需要更改），选择需要的会话，然后单击“Change Session”按钮进行更改。如果不需要进行更改，直接单击 Cancel 按钮保持默认选项。

3．自定义会话

由于某些原因，有时可能需要自己安装桌面系统。例如某些系统配置较低的用户可能更喜欢使用轻量级的桌面管理系统，如 FVWM。为了实现最好的性能，一般选择下载源码自己编译。但很可能自己编译、安装成功的 FVWM 桌面管理系统没有出现在 Sessions 选项中。其实 Sessions 选项中的内容就是/usr/share/xsessions 目录中的一个个以 desktop 为扩展名的文件。

在终端中输入“sudo gedit /usr/share/xsessions/Fvwm.desktop”，然后将加入如下代码，在 Sessions 列表中就会出现名为 FVWM 的一项。为了方便读者理解，各行的功能已写在每行前的注释中（以“#”开头的行为注释行）。

```
[Desktop Entry]
#以 UTF-8 作为默认字符集
Encoding=UTF-8
#会话的名字，任意取一个可以概括该会话的名字就行，该名字会出现在“sessions”列表（图 3-2 菜单 2）中
Name=FVWM
#会话的注释，说明本会话的功能与特点之类，可随自己的爱好输入
Comment=my fast destop manage system ,my Favorite
#执行的软件名，这个是会话的主运行软件，必须是可执行软件名
#如果该软件没有安装在/bin 或/sbin 中，还必须加上软件的完整安装路径
Exec=fvwm
Terminal=False
#TryExec 后面的内容和 Exec 行应一致
TryExec=fvwm
Type=Application
[Window Manager]
#会话是否显示在 Sessions 列表中，true 为显示
SessionManaged=true
```

登录窗口相关内容介绍就到这里。现在，开始登录工作。

如果屏幕正中的标签提示文字是 Username（用户名），意思是需要输入用户名，就在其下面的文本框中输入安装 Ubuntu 时设定的用户名，然后按下 Enter 键或 Tab 键，标签文字会变为 Password（密码），在其下文本框内输入安装时设定的密码，然后再按下 Enter 键就进行登录。两项都输入正确，系统就会将 GNOME 桌面系统展现在你的眼前。



Ubuntu 的登录是简洁到只有一个文本框，连“登录”按钮都没有提供，输入密码后直接按下 Enter 键就开始登录。这种界面设计是照顾了 Linux 用户的需求，但对于 Windows 用户来说，可能稍微有点不适应。不过这种超简洁的方式在最新版（Ubuntu 8.10）的登录窗口中已经有所改观。

3.2 认识桌面环境

现在，终于进入了 Linux 桌面系统中最重要的部分——桌面环境。Ubuntu 安装的默认桌面环境是 GNOME。GNOME 的基本概念在 2.1.5 节中已经有过介绍，这里不再赘述。本节主要介绍 GNOME 的界面元素，以及如何利用虚拟桌面提高工作效率。

3.2.1　桌面上都是些什么

进入 GNOME 桌面系统，首先显示的界面如图 3.3 所示。可以看到 GNOME 的屏幕布局有点类似于 Windows 操作系统。但与 Windows 的桌面布局不同的是，该桌面有两个面板（面板就是类似于 Windows 工具栏的东西，长条状，一般位于屏幕下部或底端，用于放置图标或显示某些状态），分别位于屏幕最上端和最下端。



图 3.3　GNOME 桌面

最上端的面板用于选择执行软件（类似于 Windows 的“开始”菜单）和显示时间、关机等操作。下端的面板用于显示当前桌面打开的应用软件和切换虚拟桌面等。为了方便起见，笔者将面板按位置进行了分区，下面按照分区分别进行介绍。

- 第 1 区是主菜单。有三个菜单项，分别是“应用程序”、“位置”、“系统”。其功能主要是运行应用程序、打开某些目录、进行系统设置等。菜单的详细内容，参见 3.2.2 节内容。
- 第 2 区是启动器。实际上就是快捷访问应用程序的图标（单击该图标可打开相应的程序，在 Windows 中有类似的功能）。系统默认的三个启动器依次是“火狐 FireFox”（系统默认的网络浏览器）、“Evolution”（系统默认的邮件管理软件）、“系统帮助”（系统功能的介绍，英文）。可以把自己常用的应用程序添加到里面，以方便打开，方法在 3.5.1 节进行介绍。
- 第 3 区内容较多。由左向右，第一个类似于计算机的图标功能是显示本机的网络连接状况，第二个键盘图标指示输入法状态，喇叭图标用来调整音量和进行音频相关设置。紧接着是日期时间显示，最后的按钮是关机键，按此按钮可以在出现的对话框中选择关机，重启，退出登录等操作。
- 第 4 区很简单，是一个“切换到桌面”的按钮。可以在窗口被应用程序遮盖时显示桌面。
- 第 5 区显示当前桌面所有打开的应用程序。单击程序的按钮可使相应的应用程序显示在桌面的最上方。
- 第 6 区先是虚拟桌面按钮（默认是两个），主要功能是切换虚拟桌面。在打开 3D 桌面特效时虚拟桌面是很重要的一部分，具体设置方法将在后面的小节里进行介绍。然后是“回收站”按钮，删除了的文件都放在里面。如果错误删除了某个文件，可以从里面找回来。

屏幕上除了这两个面板，其余的部分称为桌面。图 3.3 的桌面上显示了两个打开的应用程序，一个是文件管理器，一个是 OpenOfice.org 办公软件套件中的电子表格软件。对于熟悉

Windows 操作的用户来说，Linux 的桌面操作和 Windows 有许多相似之处。所以在稍微了解后，基本操作应该没有什么障碍。但对于这些用户而言，有些在 Windows 中已经熟悉的操作在 Linux 中却不知该如何进行。想在 Linux 中找到原来在 Windows 操作经验中的操作方法，请参见表 3.1，该表列出了 Linux 的实现方式及其对应的 Windows 实现方式。

表 3.1　Windows 用户指引

执行的操作	Linux 实现方式	Windows 实现方式
打开应用程序	第 1 区“应用程序”菜单	开始菜单
显示特定目录，比如我的文档，我的图片	第 1 区“位置”菜单	开始菜单“我的电脑”或桌面“我的电脑”图标
查找文件	第 1 区“位置”菜单，“查找”项	开始菜单中的“搜索”项
系统设置	第 1 区“系统”菜单“首选项”选项和“系统管理”选项	控制面板
快速启动项	第 2 区启动器	工具栏快速启动项
开机、关机、重启等操作	第 3 区关机按钮	开始菜单关机选项
显示桌面	第 4 区切换按钮	快速启动项中“显示桌面”按钮
虚拟桌面	第 6 区虚拟桌面	不支持
被删除文件的暂时存放地	第 6 区回收站	桌面回收站图标



除了这些操作上的相似性外，Windows 中的各种常用软件，一般来说在 Linux 中也有相应的免费软件可供使用。这些软件在后面的相关章节中将进行介绍。

3.2.2　主菜单项

主菜单项很直观，有三项内容，分别是应用程序、位置和系统。从名字就可以看出基本功能，一个用来打开应用程序，一个用来打开某些位置，一个用来进行系统设置，当然这些菜单里面的内容是很丰富的。

下面用三个表格来分别说明这三个菜单的功能。对于有些从菜单项就可以直观地看出菜单的功能的，就不在列表中说明，如“办公”菜单项中的级联菜单的内容，分别是文字处理、电子表格等，这些程序一般用户均知道其功能，就不在表中列举。如果该项菜单有什么值得注意的地方，会在其后的“备注”栏中说明一下。由于排版的原因，可能某些同一个菜单项中的级联菜单被放置在两个页面内，请读者阅读时注意。

通过“应用程序”菜单中的各项，可以打开系统中大多数的应用程序，如办公软件、网络浏览器、图形图像软件等。表 3.2 简单介绍了“应用程序”菜单中各项可实际打开的应用软件以及这些软件的功能介绍。

表 3.2 “应用程序”菜单介绍

菜单项	级联菜单	功　能	备　注
办公	略	包括文字处理，电子表格，幻灯片制作，邮件收发等几款软件	这几款软件，除了邮件软件外，均是 Openoffice.org 办公套件的一部分
附件	终端	类似于 DOS 窗口的命令行窗口，但功能强大得多	熟悉命令行操作，是提高 Linux 操作水平的必由之路

续表

菜单项	级联菜单	功　能	备　注
	密码和加密密钥	建立和管理 PGP 和 SSH 密钥，使用 PGP 和 SSH，会在很大程度上提高系统和文件的安全性	
	其他（略）		
互联网	pidgin 互联网通信程序	整合了几乎所有的即时通信软件，包括 MSN、QQ 等	
	Transmission	BT 客户端	
图像	GIMP	可以媲美 PhotoShop 的图像处理软件，支持图层、插件等高级特性	
	Xsane	可用于大多数扫描仪，扫描速度也很快	
影音	略	系统自带的影音软件其实不怎么好用，将在第 5 章介绍更好的软件	
游戏	略	一些休闲小游戏，有数独，国际象棋等	
添加/删除		提供了一个简洁的添加/删除软件的方式	

通过“位置”菜单中的各项，可以打开系统特定的目录，如类似于 Windows 系统“我的文档”、“图片收藏”等的用户目录，还可以打开系统中其他存储设备，如光驱及其他文件系统。表 3.3 简单介绍了“位置”菜单中可以打开的目录。

表 3.3 “位置”菜单介绍

菜 单 项	功　能		备　注
主文件夹	打开用户主目录及相关其他目录	打开用户主目录	
桌面		打开用户桌面目录	
文档		打开用户文档目录	
音乐		打开用户音乐文件目录	
图片		打开用户图片文件目录	
视频		打开用户视频文件目录	
计算机	打开用户系统中安装的硬件	打开硬件列表，包括硬盘、光驱、软件等	
CD/DVD 刻录机		打开光驱	
软盘驱动器		打开软驱	
10.2GB 介质		打开系统已经挂载的一个分区	这个分区由于计算机挂载的分区不同会略有不同
网络	连接到网络中的其他计算机	相当于 Windows 的“网上邻居”，查看在同一个局域网中的其他计算机供分享的内容	
连接到服务器		通过 SSH、FTP、HTTP 等方式登录到网络中其他计算机	
搜索文件	搜索文件及其他功能	以文件名，或文件所含内容来搜索计算机中的文件，支持简单的正则表达式，单击“选择更多选项”下拉框，会显示更为丰富的选项	搜索最快的方法是使用控制台命令中的 locate
最近的文档		显示最近打开的文档	如果有隐私文件，请及时删除这个列表

“系统”菜单的各项主要是用来进行系统设置，如中文输入法、屏幕分辨率等。表 3.4 简单介绍了“系统”菜单中各项可进行的系统设置功能。

表 3.4 “系统”菜单介绍

菜 单 项	级联菜单	功　能	备　注
首选项	SCIM 输入法	设置中文输入法的地方，可以配置拼音、五笔等输入法	
首选项	窗口	更改窗口行为，如让鼠标移到窗口 3 秒后窗口自动到最上层来	
	会话	更改系统启动时启动什么软件，及其当前系统中正在运行的软件	
	键盘组合键	可以定义键盘的组合键，方便打开软件。	
	屏幕分辨率	更改屏幕分辨率	
	首选应用程序	更改系统默认的打开文件方式，如默认的网络浏览器，是默认的电子邮件程序等	
	搜索和索引编制	编制索引，提高搜索速度	
	外观	更改桌面的显示，内容很多，包括窗口样式，桌面背景，文字等	
	远程桌面	连接到远程桌面	
	主菜单	自定义主菜单的内容	自由删除和添加主菜单的内容
系统管理	Hardware testing	执行硬件测试	
	登录窗口	更改登录窗口的样式及登录方式	
	服务	设置随系统自动启动的服务，有当前系统中运行的服务	
	更新管理器	设置更新系统的方式	
	软件源	设置一个速度更快的软件镜像地址	实际是编辑 /etc/apt/sources.list 文件
	授权	设置软件和用户使用系统资源的权限	
	系统监视器	显示和管理系统资源的使用情况，管理当前运行的软件，如果软件停止响应，可在这里强行关闭该软件	
	新立得软件包管理器	最主要的软件安装与管理方式	命令行方式为 apt-get
	用户和组	设置用户和组	
帮助和支持	无	获得系统帮助	是英文内容
关于 GNOME	无	显示 GNOME 版本号及开发人员，版权信息	
关于 Ubuntu	无	显示 Ubuntu 版本号及开发人员，版权信息	
退出	无	执行关机、重启、注销及挂起等系统操作	

阅读上面的表格，会发现主菜单项里面功能极其丰富。菜单项内容还会随着用户安装软件而更加丰富。可以说熟悉了主菜单，就已经熟悉了 Linux 桌面的大半内容。虽说还有其他桌面系统，如 KDE 和一些轻量级桌面，但触类旁通，熟悉了 GNOME，其他桌面系统也很容易上手。

不过，这么复杂繁多的内容，可能会让许多读者心存疑惑：“这么多，我记不住怎么办？”

没关系，这么多内容，没有几个人记得住的。只要知道这些菜单的主要功能，然后在使用时查阅一下表格就行了。有几个原则，可以帮助记忆：

- 要打开应用程序，去“应用程序”菜单；
- 要打开某个目录（即文件夹），去“位置”菜单；
- 与系统外观表现相关的功能，在“系统”|“首选项”里看看；
- 系统内在的设置，在“系统”|“系统管理”里找找。

这里只是提及了实现各项功能的位置，但具体的功能实现还有许多的技术细节。有的技术细节并不直观，对于这些内容，将在本书其他相关章节进行详细介绍。下面将介绍提高工作效率的利器：虚拟桌面。

3.2.3　使用虚拟桌面

可能许多读者从媒体报道中看到 IT 从业人员或财经界操盘手的工作台上，往往摆放着数台显示器。其中一台用来显示图表，另一台则充满文字；或者一台显示代码、另一台显示渲染出的图像之类。据说，对于这些工作，双显示器可以提高 70%的工作效率。当然，不是每个人都会在自己的工作台前放置两三台显示器。但是很多时候，却确实感觉到打开的程序、文档太多，很难找出需要的是哪一个。

面对这种困境，一个显然易见的选择是用 Alt+Tab 组合键切换。但如果同时打开四五个终端，再开一个虚拟机，还在使用一个字处理软件，再加一个网页浏览器，再加聊天软件、音乐播放软件等，按下 Alt+Tab 组合键会显示出一个长长的软件列表，用户很难选择到自己想要的程序。这时候，简单地用 Alt+Tab 组合键切换就不再是一个明智的选择了。这种问题的解决方案，就是虚拟桌面。

1．虚拟桌面的概念

虚拟桌面，顾名思义，就是给用户虚拟出一个或几个桌面用于放置打开的程序或文档。用户可以将与文本相关的程序放在虚拟桌面 1，在虚拟桌面 2 上打开图像相关的程序，把音乐播放软件放在虚拟桌面 3 上打开，以此来提高工作效率。

2．虚拟桌面的基本知识

虚拟桌面的使用很简单。在讲述之前，先明确一下概念，虚拟桌面是指 3.2.1 节中所介绍的可以放置图标、显示打开程序的桌面，而虚拟桌面按钮则是指图 3.4 右下角用来切换桌面的两个按钮。

先看看如图 3.4 所示的虚拟桌面按钮。默认有两个按钮，每个按钮代表一个桌面。橙色的为当前桌面按钮，代表当前显示出来的桌面；暗灰色的为备选的虚拟桌面按钮，代表没显示出来的哪个桌面（图 3.4 中深色按钮为当前桌面按钮，浅色按钮为备选桌面按钮）。

随便打开一些程序，可以看到当前桌面按钮发生了细微的变化。仔细观察，会发现该按钮上显示出了当前打开窗口的缩略图。图 3.4 虚拟桌面按钮上有两个按钮，其中深色的按钮里面包含图形的形状和桌面打开的窗口是一致的。按钮中深色的黑框矩形代表了“文件管理器”窗口，而在其后的矩形代表了“计算器”程序。

单击另一个“虚拟桌面按钮”，让另一个虚拟桌面成为当前桌面，可以看到该按钮成为橙色。然后显示出备选虚拟桌面中的内容。如果该桌面没有打开任何程序，则只是在桌面上显示桌面背景。此时，单击左边“虚拟桌面按钮”中的一个矩形（代表了一个程序），然后把该矩

形拖动到右边的“虚拟桌面按钮”中来，可以看到，在第 1 个虚拟桌面中打开的程序被移动到了第 2 个“虚拟桌面按钮”。这是一种在虚拟桌面间移动程序的方法。



图 3.4　虚拟桌面按钮

还有一种在虚拟桌面间移动程序的方法是在程序的标题栏上单击鼠标右键，在弹出的菜单中选择“移动到右侧（或左侧，取决于程序所在的位置）的工作区”项，此处菜单中所谓的“工作区”就是虚拟桌面。然后此程序就从当前桌面上消失了，别担心，看看虚拟桌面按钮的图像，就会发现该程序已经被移动到了另一个桌面。

3．增加虚拟桌面的数量

虚拟桌面默认有两个，可能不能满足特定用户的需求。增加虚拟桌面的方法如下：

（1）右键单击虚拟桌面按钮，在出现的菜单中选择“首选项”，会弹出一个对话框；

（2）在“Number of workspaces”（工作区的数量）微调框中单击向上的微调按钮，就会增加虚拟桌面的数目。也可直接在微调框中输入想要的数目。

4．虚拟桌面的组织技巧

通过以上的介绍，读者应该对虚拟桌面有了初步的了解。一般而言，虚拟桌面使用前，应该先评估当前操作，将功能分块，如文字处理一块，图像处理一块。然后让文字处理相关内容显示在桌面 1、图像处理相关程序显示在桌面 2 等，这样会让各种操作方便许多。习惯虚拟桌面后，你会离不开这种操作方式。

5．虚拟桌面的组合键

为了提高效率，当然要提及一下组合键。下面再来介绍虚拟桌面的组合键操作。如果正在 gedit 中处理文本，觉得当前桌面过于拥挤，想移动 gedit 到右边的虚拟桌面，不用再动用鼠标，只需按下 Ctrl+Alt+Shift+右方向键即可。移动任何一个程序都可在该程序内用此 4 个组合键来进行。

6．设置虚拟桌面切换的视觉效果

如果在图 3.5 所示“视觉效果”标签页中选择了“正常”或“扩展”单选按钮，就开启了高级桌面效果模式。开启高级效果模式后，直接在桌面内拖动程序超过窗口的边界，则该程序自动进入到该边界外的另一个桌面。还以 gedit 为例，如果当前桌面是第 1 个虚拟桌面，将 gedit

拖动到当前桌面的右边界处，然后继续向右拖动，可以在 2 号“虚拟桌面按钮”中看到有一个矩形框慢慢进入。



系统默认没有启动视觉效果，所以默认效果下，程序拖到窗口的边界，不会进入下一个虚拟桌面。

如果开启了 3D 桌面效果，虚拟桌面的切换就会变得炫目之极，如以 3 维立方体方式切换、以水中倒影方式切换等，再搭配上自己定义的组合键，似乎那些繁杂枯燥的工作都会变得美丽起来。微软也推出了自己的虚拟桌面软件，不过切换的方便程度和切换的炫目效果比起 Linux 来，还是有相当的差距的。如何开启 3D 桌面效果，同样参见 3.3 节相关内容。很多时候，会觉得系统默认的显示效果还不够美丽，3.3 节介绍的内容主要就是将系统装扮得更漂亮。

3.3　美化系统

美化系统包括许多方面，如桌面系统元素的美化、选择美观的字体、开启 3D 特效等。最基础的工作就是桌面系统元素的美化，通过桌面系统元素的美化，可以充分地展示使用者的个性及审美水平。本节就介绍这方面的内容。

3.3.1　更改桌面背景

桌面系统，就外观方面来说，很重要的元素之一就是桌面背景。一幅好的桌面背景可以让人心旷神怡，或者至少可以提高工作效率。所以，许多用户使用计算机的第一个步骤就是更换桌面背景。而且，更换桌面在大多数桌面系统中都是很容易的事，在 Ubuntu 中也不例外。具体步骤如下。

（1）右击桌面空白处，在弹出的快捷菜单中选择“更改桌面背景”命令，出现的对话框如图 3.5 所示。可以看到，Linux 把背景图片称为“壁纸”。

（2）在“样式”下拉框中可以选择壁纸铺设的方式，一般选择“居中”或“填充屏幕”项。

（3）图 3.3 中在列表框里有 4 张壁纸，选择任意一张，就可以将其设为当前壁纸。

如果要把自己收集的美丽图片作为背景壁纸，单击“添加”按钮，在弹出的对话框中定位到自己的图片，选定图片后单击“确定”按钮，该图片就会出现在列表框中。选项列表框中的图片就可将其设定为当前桌面的背景。

如果不想让某张图片出现在列表框中，在列表框中单击选中该图片，然后单击“删除”按钮，该图片就会在列表框中消失。但该图片其实并没有从硬盘中删除，而只是不显示在该列表中而已，以后还可以将其加入列表框中。

如果系统中没有合适的图片作为壁纸，可以进入如下网址搜索，一定会有收获。

- http://www.desk9.com；
- http://www.linuxwallpapers.org；
- http://www.desktopnexup.com；
- http://www.flickr.com。



图 3.5　更改桌面背景

3.3.2　美化登录方式

可以美化的登录方式包括 Grub 启动选单、splash 开机画面、GDM 登录窗口。默认情况下，这些项目都是千篇一律的样式，经常面对这些，无疑会感觉到有点厌倦，而且也显示不出来个性。下面分别介绍如何更改登录方式的界面。

1．安装启动管理器

安装启动管理器后就可以进行启动项目的设置，如启动时的外观、启动时的安全性等。安装步骤如下所示。

（1）依次选择菜单“系统”|“系统管理”|“新立得软件包管理器”命令，弹出对话框。

（2）在对话框中单击“搜索”工具按钮，弹出对话框。

（3）在对话框“搜索”文本框中输入 startupmanager，单击“搜索”按钮进行搜索。



图 3.6　启动管理器

（4）在搜索结果列表框中用鼠标右键单击 startupmanager，在弹出的快捷菜单中选择“标记以便安装”命令。

（5）单击“应用”按钮，开始下载并安装。

2．让 Grub 启动选单拥有背景图片

安装了启动管理器后，就可以进行相关设置了。本次进行启动选单的美化工作。步骤如下所示。

（1）依次选择“系统”|“系统管理”|“启动管理器”命令，打开启动管理器。

（2）选择“外观”标签，打开如图 3.6 所示对话框。

（3）单击“管理启动加载器主题”按钮，弹出对话框。

（4）在其中单击“添加”按钮，添加要显示在 Grub 中的背景图片。现在会回到如图 3.6 所示界面。

（5）在“Grub 背景图片”下拉列表框中选择刚才添加的图片，然后选择“启动加载器菜单采用背景图片”复选框，这样该图片就会显示在 Grub 启动管理器的界面中。

3．更改 splash 开机画面

开机画面的设置同更改 Grub 背景图片很相似，同样是在启动管理器中进行的，参照上述

步骤打开启动管理器，然后进行如下步骤即可。

（1）在如图 3.6 所示界面中，单击“管理 Usplash 主题”按钮。

（2）在弹出的对话框中单击“添加”按钮添加 splash 主题。

（3）回到如图 3.6 所示界面后，在“Usplash 主题”下拉列表框中选择相应的主题即可更改 splash 画面。

4．GDM 登录界面

选择“系统”|“系统管理”|“登录窗口”命令，打开登录窗口管理器。选择“本地”标签，选择相应的主题就更改了 GDM 登录界面的外观。

3.3.3　使用其他主题

Ubuntu 系统自带了几个主题，默认主题是 Human，意思是人人都爱的风格。但其实个人喜欢的是“薄雾”主题，因为其关闭按钮比较大。这些主题可以轻易改变，而且 Ubuntu 提供了比 Windows 更为丰富和深入的设置细节来更改计算机的外观。Ubuntu 和 Windows 不一样的是，前者的主题不包括字体、桌面背景项。如果要调整这两项，应该自己手动设置。

1．使用系统已经安装的主题样式

更改主题的方法很简单，按如下步骤。

（1）选择“系统”|“首选项”|“外观”命令，或者在桌面上用鼠标右键单击，在弹出的快捷菜单中选择“更改桌面背景”命令，再选择“主题”标签，就打开了主题设置对话框，其界面如图 3.7 所示。



图 3.7　自定义主题

（2）单击主题列表框中的任一款主题，则立即就可以看到主题已经被应用。在“外观首选项”对话框，已经安装了十几款主题，可以自由选用。

（3）单击“主题”标签页里“自定义”按钮，弹出“自定义主题“对话框，可以设置主题的一些细节，如边框、目录图标等。在该对话框里，可以详细定义 GNOME 桌面的各个细节，如各种控件的样式、控件的颜色、边框的形状、图标、指针等项。

2．安装软件库中自带的主题

尽管 Ubuntu 提供了如此丰富的外观特性供选择，但一定会有用户不满足于此。显然，Ubuntu

提供了更多的主题供这些用户选择。用户可以下载完整的主题或者仅仅下载某个细节特性，如边框的样子。下载 Ubuntu 源文件中附带的完整主题的方法如下。

（1）依次选择主菜单“系统”|“系统管理”|“新立得软件包管理器”命令，弹出“新立得软件包管理器”对话框。

（2）在“新立得软件包管理器”界面中单击工具栏里“搜索”按钮，弹出“查找”对话框。

（3）在“搜索”文本框内输入“gtk2-engines”（Ubuntu 把主题文件称为“gtk2-engines”），单击“搜索”按钮。

（4）“新立得软件包管理器”会列出软件库中的所有主题，从名字上选择感觉不错的主题进行安装。这些主题无法进行预览，只有简单的说明文字，只有下载后才能知道其特性。可以根据自己的喜好进行安装。

（5）被安装主题出现在图 3.7 中的列表框中，可以选择使用。

3．从互联网下载主题

如果对于系统中附带的主题觉得不够满意，也可以在互联网上下载主题文件。可以去 GNOME 官方艺术网站 http://art.gnome.org 寻找主题，或去专门为 GNOME 准备主题的 http://www.gnome-look.org/网站下载主题，这两个网站均有许多主题。前者包括桌面背景及细节方面的主题样式，后者内容稍微丰富些，有许多方面的主题。笔者推荐后者。这些网站虽然使用的是英文，但主题主要是看界面。这两个网站均对每一个主题提供了图片，单击图片就可放大查看。可见，基本上语言不是问题。

这两个网站上均提供了许多图片，选择看起来不错的主题文件（一般包含“theme”字样的为主题文件），单击图片，会打开新的一页，该页面会在醒目的位置提供下载的链接（英文是“download”），单击该“download”就可以下载。一般情况，该文件是以.tar.gz 为扩展名的压缩文件。

主题文件的安装方法有两种：一是下载后，不用解压，单击如图 3.7 所示的“安装”按钮，找到下载的文件位置就可以进行安装，安装完成就可以看到主题已经出现在列表框中；二是选择如图 3.7 所示的“主题”标签，将下载的主题文件拖进去，就自动安装了。

示例：苹果公司（Apple Inc）向来以卓越的工业设计享誉业内，其外观样式（无论是其操作系统 OS X，还是其 iPod 等硬件设备）也有许多人拥趸者。在 Ubuntu 众多的主题中，也有一款是模仿 Mac 风格的。下面就以该主题为例说明安装主题的方法。

（1）在 http://www.gnome-look.org/content/show.php/OSX-Tiger+theme?content=56577 下载该主题包。

（2）下载完成后，右击桌面，选择“更改桌面背景”命令，弹出如图 3.7 所示对话框。

（3）在弹出的对话框单击“主题”标签，切换到“主题”标签页。

（4）把下载的主题包拖进该对话框，系统会提示是否安装，单击“安装”按钮，即可完成主题的安装。安装完成后，就可以看到被安装的主题出现在图 3.7 中的列表框中，可以选择使用。



安装的网上下载的主题，其名称为“自定义”。而且运用该主题后，感觉还是和 Mac 风格有所差异。所以应该再对其进行设置。单击图 3.7 中“自定义”按钮，打开设置对话框。该对话框中显示了几个标签页，每个标签页中各有一个列表框。在“窗口边框”标签页中选择“Glossy”项，在“图标”标签页中选择“Crystal SVG”项，在“指针”标签页中选择“DMZ White”项。这样看上去就与 Mac 风格比较类似了。

3.3.4　美化系统字体

字体设置直接关系到使用时的感受，单击菜单时，要看菜单上的字体；打开对话框时，要看各种控件上的字体；文本编辑时要看字体。如果选择了看起来不舒服的字体，无疑是对宝贵的视力的虐待。由于中文字体笔画复杂，而且数量巨大，所以制作中文字体极其耗费精力与时间，这直接导致了自由字体极少。这种情况下，用户不得不在 Linux 中使用版权不明的字体来美化系统。

1．文泉驿字体简介

文泉驿的出现改变了中文自由字体稀缺的情况。其创始人是房骞骞，在使用 Linux 时，感受到了“不完备和模糊的汉字显示带来的不便”，所以产生了使用 wiki 的协同编辑方式来开发开源点阵字体的想法。由于字体制作工作量大，同时需要技术性和艺术性，其进展并不快。不过，现在文泉驿的字体开发具有很低的门槛。其网站主页也在鼓励大家的参与。该项目的主页是 http://wenq.org/。有精力参与的朋友不妨打开该网站，看看是否有自己可以参与的工作，大家共同努力让这些自由字体健康、快速地发展。

2．文泉驿字体的安装

Ubuntu 8.04 版默认使用文泉驿正黑字体。该字体是第一个完整包含简体与繁体中文的开放源码黑体字，还内嵌了文泉驿点阵字，显示效果很不错。文泉驿系列包含许多字体，但默认安装的字体比较少。如果需要更多的字体，可以在网站 http://wenq.org/下载字体，也可以在新立得软件管理器中搜索 wqy 进行安装，或者在终端中输入“sudo apt-get install xfonts-wqy”进行安装。

3．系统字体设置

Ubuntu 系统字体设置包括几个部分，最重要的是系统字体的设置。系统字体是全局性的，包括菜单、对话框、标题栏等。也就是说，如果设置了系统字体，则系统中所有的对话框、菜单、标题栏均按照该设定进行显示。设置系统字体的步骤如下所示。

（1）在桌面空白处右击鼠标，在弹出的对话框中选择“更改桌面背景”命令，会出现“外观首选项”对话框。

（2）选择“字体”标签，切换到字体设置页，如图 3.8 所示。



图 3.8　字体选择

等宽字体可以选用 DejaVu Sans Mono。据说该字体派生自 Bitstream Vera Sans Mono。不过这两款字体相当相像，显示效果均很美观。除了等宽字体外，其余字体均可选择“文泉驿正黑”（WenQuanYi zenhei）字体。

渲染方式共有 4 个选项，在不同的系统上会有不同的效果。可以自己观察，选择在自己机器上显示效果最好的一个。液晶显示器一般可以选择“次像素平滑”单选按钮。如果对于细节还是不够满意，可以单击“细节”按钮，进行一些调整。

这些选项里的其他条目比较好理解，但等宽字体就有点费解，很难从字面上看出来其作用与功能。下面简略介绍等宽字体的知识。

4．等宽字体

等宽字体（Monospace Font）是指字符宽度相同的计算机字体。与此相对，字符宽度不尽相同的计算机字体称为比例字体。在传统西文印刷中，比例字体可以提高单词的可读性。早期的计算机屏幕显示、打字机等由于技术的局限，无法进行字母宽度的比例调整，因此将每个字符都制作成一样的宽度，从而形成了等宽字体。现在虽然不存在这种问题，但对于终端而言，设置一款美观的等宽字体，仍然可以改善终端的显示效果。如果选择比例字体用于终端显示，可能会出现一些较窄的字母（如 i、j）挤在一起的情况。

5．衬线字体与非衬线字体

在字体设置中，衬线字体（serif）与非衬线字体（sans serif）也是常见选项。衬线字体的意思是字体的笔画有着特殊的装饰。其实，这对于中国人来说很好理解，因为大多数的毛笔字体均是衬线字体，比如颜体、柳体等。而非衬线字体是指笔画的始末均没有特殊的装饰，是一样粗细的，如常见的黑体字就是非衬线字体。



图 3.9　非衬线字体与衬线字体

如图 3.9 所示，“驿”就是非衬线字体，而“境”则是衬线字体。常见的字体中，宋体、楷体均是衬线字体，而幼园、黑体等则是非衬线字体。Ubuntu8.04 默认的字体文泉驿正黑就是非衬线字体。

3.4　开启酷炫的桌面特效

使用过 Vista 的朋友可能会被 Vista 的 3D 特效所吸引，但其实 Linux 中也可轻易实现类似的特效。而且特效的种类、显示的美观程度更是有过之而无不及。Ubuntu 中可以实现的桌面 3D 特效种类有许多，如 Beryl、AiGLX、XGL、Compiz Fusion 等。从 Ubuntu 7.10 开始，系统默认的 3D 特效是 Compiz Fusion。本节以 Compiz Fusion 为例介绍 3D 桌面的安装及设置。

3.4.1　安装显卡驱动

要在系统中开启酷炫的桌面特效，并使其流畅运行，显卡的支持是一个必要的前提。鉴于 Linux 系统对于一些显卡支持程度不够的情况，在本节还将介绍安装显卡驱动的方法。

根据 Ubuntu 官方声明，称配置 PIII 866 的计算机也可以成功运行 3D 桌面。但如果要支持更多的特效、追求使用的流畅程度，配置也不能太低。CPU 主频在 2GB 左右，内存容量 1GB 应该可以应付大多数的 3D 特效。当然，这样的配置对于现在的计算机应该不是什么问题了。

一般的显卡厂商，只是努力地与 Windows 兼容，努力去通过微软的论证，对于 Linux 的支持就差多了。而由于显卡的硬件等各方面的代码并未开源，所以 Linux 无法获得底层信息，驱动程序就不一定能获得最佳的性能。例如，一般的 Linux 系统对于 ATI 系列的显卡支持程度就不怎么高，原来安装驱动也是一件很复杂和容易出错的工作。但现在如果使用 NVIDA 或 ATI 显卡（这两家厂商的产品占据了市场的大部分份额），就可以使用一款叫 envyng 的软件安装更适合机器的驱动程序，以完全发挥显卡的性能。安装与设置显卡的步骤如下。

（1）在新立得软件包管理器里搜索 envyng-gtk，找到后直接双击或者用鼠标右键单击选择安装，然后选择“应用”将其安装。也可以使用命令行方式，命令行安装则只需在 Shell 中输入“sudo apt-get install envyng-gtk”即可。

（2）选择“应用程序”|“系统工具”|“envyng”启动设置对话框。

（3）启动后会出现一个对话框，提示用户选择显卡类型。按计算机中所实际安装显卡类型，选择 ATI 或者 NVDIA。

（4）在右边选择默认的“Automatic Hardware Detection”（自动检测硬件安装）。

（5）单击“应用”按钮，即可自动下载安装官方驱动程序了。

安装 Compiz Fusion

为了启动 3D 特效，需要安装一些必需的软件，包括提供特效的主程序，以及提供桌面装饰的程序，还有一个方便用户管理与设置的程序。安装这些软件的方法很简单，在终端中输入下列命令即可：

```
sudo apt-get install compizconfig-settings-manager emerald fusion-icon
```



compizconfig-settings-manager 是用来提供 3D 桌面特效的主程序，而 emerald 则是提供窗口装饰的程序，fusion-icon 则是一个桌面栏中的图标。

安装完成后，选择“应用程序”|“系统工具”|“Compiz Fusion icon”命令，就开启了 Compiz Fusion。



在未进行设置前，桌面没有什么改变。所有的桌面特效均需进行设置才能开启。

3.4.2　设置 3D 立方体效果

Compiz Fusion 安装完成后，在屏幕的右上角，会出现一个蓝色的立方体图标。这就是 Compiz Fusion icon。右击该图标，在弹出的菜单中选择“Settings Manager”命令，就会弹出特效设置窗口。其界面如图 3.10 所示。

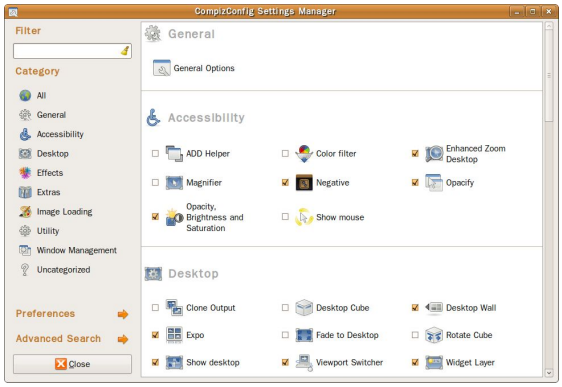


图 3.10　Compiz Fusion 设置界面

不过，在开始设置前，首先需要将虚拟桌面数量更改为至少 4 个。可以先参照 3.2.3 节内容将虚拟桌面数量设置为 4 个，然后参照如下步骤进行设置，就可打开 3D 立方体效果。

（1）选择“Desktop”（桌面特效）选项卡中“Desktop Cube”（桌面立方体）和“Rotate Cube”（旋转立方体复选框）。

（2）选择“Effects（特效）选项卡中的“3D 窗口”复选框，如果出现“开启 3D 桌面会关闭桌面背景”之类的提示，单击“确认”按钮即可。

（3）在桌面空白处右击，在弹出的快捷菜单中选择“更改桌面背景”命令，弹出“外观首选项”对话框。

（4）在弹出的对话框中选择“视觉效果”标签，选择“扩展”单选按钮。

现在就开启了 3D 桌面特效。按下 Ctrl＋Alt＋鼠标左键，然后移动鼠标就可以看到类似如图 3.11 所示的 3D 桌面特效了。



图 3.11　3D 立方体效果桌面特效



Compiz Fusion 有许多的特效设置功能，如实现水中倒影效果、桌面的火焰特效等。这些设置一般来说是比较简单的。不过，该设置程序的汉化程度不是很好，许多选项依然是英文界面，好在这些单词并不是很难，对于稍有些英文基础的读者来说应该不难理解。

3.5　定制系统以适合自己的需要

Linux 提供了丰富的可设置项，可以用来设置系统的各个层面。从系统底层的配置到访问软件的方式均包含在其中。3.4 节介绍了美化系统的内容。本节主要介绍进行系统设置以快速访问软件的方法，以及一些设置笔记本电脑屏幕亮度的方法等内容。

3.5.1　把自己喜爱的软件添加到面板

每个人都有许多自己习惯使用的软件。但很可能你经常使用的软件并没有被系统放在容易访问的位置。这样每次使用时不得不从几级菜单中寻找，比较麻烦。一个简单易行的方案是把这些常用软件加入启动器中，使用起来就容易多了。将软件加入启动器有几种方法。下面依次进行介绍。不过，在进行这项工作之前，应该先做点准备工作。方法是在桌面顶端的面板上右击，在弹出的快捷菜单中选择“添加到面板”选项，出现如图 3.12 所示的对话框。

1．添加系统提供的程序

注意列表栏中细细的灰线，这条线是一个分界线，区分出高手与新手。线下是新手专区，

系统放置了一些认为可能有用的程序，只需双击，这些程序就会出现在面板的启动器区。当然，里面有一些有用或有趣的程序。

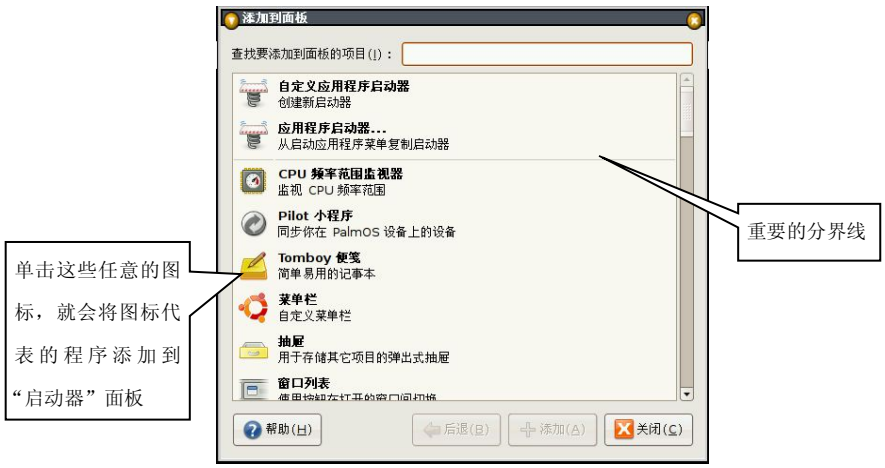


图 3.12　添加到面板

2．添加主菜单中的程序

线上自然是高手区了，先看第二项“应用程序启动器”。在列表框中双击该行，弹出如图 3.13 所示的对话框。可以在“查找要添加到面板的项目”文本框中输入应用程序名，然后按下 Enter 键查找。如果记不清楚名字，可单击列表框中每行前的三角符号展开各项，寻找自己熟悉的程序。找到程序后，双击程序，则该程序会出现在启动器栏中。



许多用户可能对于如图 3.13 所示列表框的内容有似曾相识的感觉。因为其中的内容就是主菜单“应用程序”项里的内容。而图 3.13 其实就是提供了一个便捷的途径，以方便用户把“应用程序”菜单中的程序放到“启动器”中。

3．添加其他程序

图 3.12 列表框中第一项“自定义应用程序启动器”，难度是最高的。如果成功完成第 1 项的操作，说明用户对 Linux 系统已经有了比较深入的认识。添加其他程序的方法是先双击“自定义应用程序启动器”，出现如图 3.14 所示的对话框。然后按下列各项功能说明，进行相应的设置即可。下面是如图 3.14 所示各项内容的说明。

- “类型”是指要放进启动器栏中程序的类型，选项有 3 个：应用程序、终端中的应用程序、位置。应用程序是指仅拥有图形界面的程序（区别于字符界面的程序），如 OpenOffice.org，GIMP 等。Linux 有许多只能在终端中运行的程序，这些程序只拥有字符界面。这种程序就应该属于“终端中的应用程序”。如果选择了“位置”选项，则启动器打开的就是一个目录。
- “名称”栏填入用户给程序取的名字，如“强大的 GVIM”，这个名字是可以随便取的。
- “命令”代表可执行程序名，该项比较重要，不能随便填。如果要放入启动器栏的程序放在“/bin”目录（或其他可被系统搜索的路径）中，则可直接填写程序名。否则就要填写完整路径，内容可能为“～/software/scilab5.0/bin/scilab”。如果不记得确切的路径可以单击“命令”后面的“浏览”按钮，在图形界面中查找程序所在的位置。



图 3.13 “添加到面板”之“应用程序启动器”



应用程序是否安装在“/bin”目录的检验方法：打开终端，在其中输入应用程序的名字的一部分，如 scil。再按下 Tab 键，如果终端自动补全了这个程序名 scilab，则 scilab 安装在“/bin”目录。否则安装在其他目录，需要输入完整路径。



图 3.14 “添加到面板”之“自定义应用程序启动器”

- “注释”是对此程序的附加说明，可有可无。
- 左上方有一个较大的正方形按钮，其上显示了一个螺丝图像，该图像是将要在“启动器”栏显示的图标。单击此按钮会打开文件浏览器，可以找一个自己喜欢的图像替换这个螺丝。系统的图标在“/usr/share/icons”目录，而且在一般情况下，软件会在自己的安装目录里提供一些图标，可以去找找看。



这个图标很重要，直接关系到使用时的方便程度。想象一下，如果在启动器栏添加了 8 个自己喜欢的程序，用的都是这个螺丝图标。那么，当启动器上并排出现 8 个一模一样图标的时候，是很难选择想要打开的一个的。所以，选择一个合适的图标很重要。

填好各项后，单击“确定”按钮，完成此操作。所添加的程序就会显示在启动器栏。鼠标放在新添加程序的图标上，会显示“名称”和“注释”文本框中输入的内容。如果各项设置正确，单击鼠标，就会打开相应的程序。

3.5.2　定制主菜单

有时候，一个程序会偶尔用到，虽然使用并不频繁，但要找到这个软件的位置却是件比较烦琐的事，往往要到很深的目录里面去找，如/home/westnorth/software/scilab/ src/bin/scilab 或类似的目录。如果将其放入启动器中，就会浪费启动器很有限的空间。这时，有一个替代方案可供选择，即将其放入主菜单中去。

将程序放入主菜单中，至少有两个好处：一是方便访问，因为打开主菜单绝对比从 4 级目录中找一个文件要快捷，而且对于普通用户来说，很可能在使用时忘记软件的位置；二是可以分类放置，比如，可以将 maxima、scilab 等数学软件放入“教育”类别的菜单项中。下面以一个例子来说明定制主菜单的优越性。

一个名叫 phun 的软件，是一个物理量模拟软件，它是美国一个研究生的毕业设计。该软件给每一个物体赋予物理特性，如重力、弹性等，是一个很不错的科普软件。但开发者并未提供 deb 包，只提供了编译好的可执行程序供下载。也就是说，该程序下载后不会自动出现在菜单栏中，需要自行添加。下载该软件及添加其到主菜单的步骤如下。

（1）从 www.phun.at 下载，得到的文件类似于这样 Phun_beta_4_11_linux32.tar.bz2。

（2）用“tar zxf Phun_beta_4_11_linux32.tar.bz2”将其解压，得到一个目录 Phun，该目录中的 phun.bin 是可执行程序。

（3）现在 Phun 的位置可能位于/home/westnorth/software/Phun/（这依赖于下载解压的目录）。以下的步骤是将其加入主菜单的过程。

（4）在主菜单上右击，在弹出的快捷菜单中选择“编辑菜单”命令。

（5）出现“主菜单”对话框，如图 3.15 所示。本次需加入的软件属于教育类，所以在左边“菜单”列表框中选择“教育”选项。

（6）再单击右边“新建项目”按钮，出现熟悉的“创建启动器”对话框。

（7）在“创建启动器”对话框填入适当的内容，单击“确定”按钮。

（8）现在检查一下操作结果。打开“应用程序”主菜单，在“教育”菜单项中就有了 Phun 命令，选择该命令，则打开 Phun 软件。



图 3.15　编辑主菜单

3.5.3 在桌面上显示或隐藏图标

习惯了 Windows 桌面的图标，如“我的电脑”、“回收站”、“我的文档”，但 Ubuntu 的桌面却空空荡荡。当然，Ubuntu 中也可以方便地将这些相应的图标显示在桌面上。具体步骤如下所示。

（1）在控制台输入 gconf-editor 打开“配置编辑器”，其界面如图 3.16 所示。



图 3.16 配置编辑器

（2）依次选择 apps | nautilus | desktop（就是把这些项前面的向右的三角箭头变为向下的箭头）命令，打开控制桌面显示的项目，如图 3.16 所示右侧列表框所示。如果将项目选中，则该项目在桌面显示，否则不显示。

表 3.5 列出了可以用配置编辑器控制的桌面图标显示项的详细说明。知道各项的功能，就可以按照自己的需求选择想要在桌面显示的项目，并选中项目前面的复选框。

表 3.5 配置编辑器桌面图标显示项及其功能说明

名　称	说　明
computer_icon_visible	在桌面上显示计算机图标
home_icon_visible	在桌面上显示主文件夹图标
network_icon_visible	在桌面上显示网络服务器图标
rash_icon_visible	在桌面上显示回收站图标
volumes_visible	在桌面上显示已挂载的卷

如果要显示回收站，则选择 rash_icon_visible 复选框。然后，这些图标就会显示在桌面上。配置编辑器号称是 Linux 的注册表编辑器，实际上是把通过文本修改的选项自动化了。

当然配置编辑器功能还远不至此，读者可以打开其中各项，查看其说明，说不定会有特别的发现。下面是一些使用技巧。

- 去掉“Bug 报告”：有时候在 Gnome 下难免会碰到应用程序崩溃，这时 Bug 报告就会出来。假如你觉得比较讨厌，就可以通过配置编辑器去掉它。依次选择 apps|bug-buddy 命令，取消选中右侧的 run_on_crash 复选框，这些 Bug 报告就再也不会跑出来烦人了。
- 让截图也有简单的特效：依次选择 apps | gnome-screenshot 命令，将 boder_effect 的值改

一下。默认是 none，改成 shadow 就是有阴影效果了；如果改成 black-line，则截图周围就有黑线。

- 限制用户可以使用的功能：在配置编辑器中，有 4 个用户功能的禁用选项，分别是 Disable Command Line（禁用命令行）、Disable Print Setup（禁用打印设置）、Disable Printing（禁用打印）、Disable Save to Disk（禁用保存到磁盘功能，也就是说，用户所做的任何更改均不允许保存到磁盘）。在左面板中，依次选择 Desktop|Gnome|Lockdown 就可以找到这 4 项设置。当使用其中任何一个选项时，不要忘记把它们设为强制选项，否则用户就有权利把它们改过来。

3.5.4　调节笔记本电脑的屏幕亮度

对于笔记本电脑用户而言，使用过程中会有许多与台式机不同的体验。其中很重要一个就是屏幕亮度的问题。由于台式机的屏幕带有 OSD 按键，可以轻易对其进行亮度等参数进行调整，但笔记本电脑的屏幕亮度需要使用键盘的功能键来实现。一般笔记本可用按 Fn+F7 组合键或按 Fn+F8 组合键来增减亮度（在调节的键帽上印有太阳标志，有“＋”号为增加亮度，有“－”号为降低亮度）。ThinkPad 系列一般可用按 Fn+Home 组合键和按 Fn+End 组合键来调节。

但这样的组合键有时在 Ubuntu 中并没有效果。有时会出现按下组合键却没有调节菜单出现，有时出现调节菜单但实际屏幕亮度并没有任何变化。下面介绍几种方法可以帮助笔记本用户调节屏幕亮度。

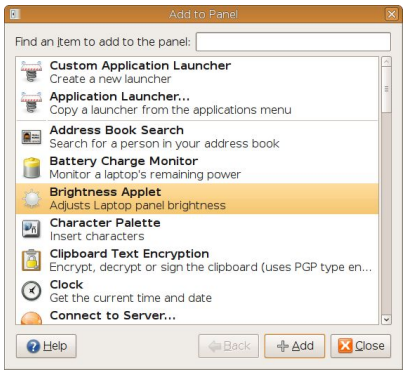


图 3.17　亮度调节插件

1．使用快捷按钮进行亮度调节

在屏幕顶端的面板空白处右击，在弹出的快捷菜单中选择“添加到面板”命令，弹出如图 3.17 所示的对话框。在列表栏中选择“Brightness Applet（亮度调节）”选项，单击“Add”按钮，将其添加到顶端面板中。

这样，面板中就会出现一个亮度调节的图标，单击其可打开亮度调节滑块。向上拖动滑块，屏幕变亮；向下拖动，则屏幕变暗。ThinkPad 系列笔记本可以在“系统”|“首选项”|“电源管理”中设置屏幕亮度。

2．修改配置文件调节亮度

当用第一项所述的几种方式都无法调整屏幕亮度的时候，则可以采用直接修改配置文件 laptop-mode.conf 的方法来设置一个最适合自己的亮度。具体方法如下所示。

（1）用“cat /proc/acpi/video/VID/LCD/brightness”命令检测该文件是否存在，若显示“文件不存在”，则进行第 3 步。

（2）记录下第 1 步的给出的几个数值，然后到第 4 步。

（3）用“cat /sys/class/backlight/’按 Tab’/brightness”命令检测该文件是否存在。注意在 backlight/目录后应该按 Tab 键让 Shell 自动补全该命令，然后再输入 brightness 即可。文件若存在，会给出一个数值。这个数值是屏幕亮度的最大值，如笔者的计算机该数值为 7。

（4）使用“sudo gedit /etc/laptop-mode/laptop-mode.conf”命令以超级用户身份打开设置文件。

（5）找到下面的代码，并进行替换。

```
ENABLE_LAPTOP_MODE_ON_AC=0
```

这一行的意思是在使用交流供电时使用 laptop-mode 模式进行亮度控制。默认是 0，意思是不使用该模式。将这一行修改为 ENABLE_LAPTOP_MODE_ON_AC=1，意思是交流供电时使用 laptop 模式控制。

（6）再寻找下述代码，进行修改。

```
CONTROL_BRIGHTNESS=0
```

这一行的意思是默认不使用本脚本进行亮度控制，将其更改为 CONTROL_ BRIGHTNESS=1，使用 laptop mode 进行亮度控制。但此时 laptop mode 还未打开。

（7）系统启动时设置屏幕亮度的默认值是由以下几行代码决定的：

```
BATT_BRIGHTNESS_COMMAND="echo 1"
LM_AC_BRIGHTNESS_COMMAND="echo 1"
NOLM_AC_BRIGHTNESS_COMMAND="echo 1"
BRIGHTNESS_OUTPUT="/sys/class/backlight/acpi_video0/brightness"
```

这几行命令后的数值，是由前面第 2 步或第 3 步决定的。如果第 1 步给出一个数值列表，则 echo 后的数值只能取这些列表中的某一个。而如果第 3 步列出了一个最大值，则 echo 后的数值可以取 0 至该数值之间的任意一个数字。如笔者系统是第 3 步得出了最大值 7，就取 0 到 7 中的任意一个数字均可。经过这些设置，笔记本屏幕亮度设置的问题就应该可以解决了。



修改配置文件调节屏幕亮度的方法只适用于 8.04 版，而 8.10 版修改了屏幕亮度的配置文件，故已无法使用该方法进行配置了。

3.6 X Window 系统

前面介绍了 Linux 的桌面系统的一些设置和技巧。但要真正了解 Linux 的图形界面系统，只知道这些已经隐藏了细节的桌面系统并不够，还需了解 X Window 系统。X Window 系统才是 Liunx 图形界面的灵魂，也是事实上的工业标准，同时也是 Linux 各发行版图形界面的基础。Linux 平台上占统治地位的 GNOME 和 KDE 桌面系统均构建在 X Window 系统的基础之上。本节将介绍 X Window 系统的基本知识，以及与 X Window 系统密切相关的字体系统。

3.6.1 简介

X Window 系统，一般也称为 X11 或 X，是负责接收用户的输入（至于输入的内容、执行何种命令则不是其职责所在），同时负责将计算机输出的信息显示出来的一套系统。X Window 系统的功能有绘制窗口、响应鼠标键盘的操作等。

简单来说，屏幕上显示的窗口中的内容（不包括标题栏及各种按钮），以及计算机接受的键盘输入信息（如在终端中输入的命令，文本处理程序中输入的文字内容）、鼠标在窗口里的移动，都是 X Window 系统在发挥作用。Linus 开发的 Linux 只是一个操作系统的核心，要与用户进行交互、接收用户的操作、显示文本或图形等就要使用 X Window 系统。

X Window 系统是基于网络协议的以位图显示的图形界面系统。X 采用 C/S 架构，也就是客户/服务器模式，由一个服务器负责与多个客户通信。服务器接收客户对于图形输出的请求，并且接收用户的输入（鼠标、键盘等），并且将这些输出到屏幕。具体结构如图 3.18 所示。

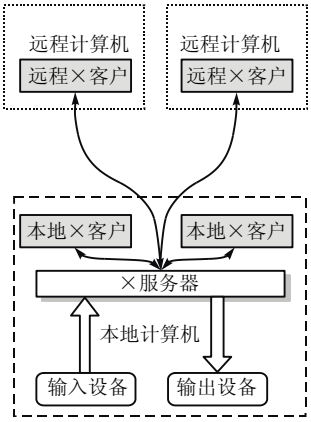


图 3.18　X Window 结构图

要注意的是图中的 X 服务器。X 服务器是指能接收指令并输出信息的设备。这个设备可以是一个软件、可以是系统的一个服务、也可能是一个硬件设备。运行 X 客户的计算机不仅限于本地机器，对于远程计算机也一样支持。同样，运行 X 客户的计算机也可以和运行 X 服务器的计算机属于不同的架构，如 X 服务器运行于 Linux 中，而 X 客户的显示请求发自 Mac OS 中，这对于 X 来说没有什么不同。

X 里客户/服务器与一般概念上的客户/服务器是有很大区别的。一般概念中，如果使用本机（A）的网络浏览器浏览远程主机（B）上的网页，因为 A 请求发送网页内容，而 B 就发送了网页，所以称本机 A 为客户端，而称远程主机 B 为服务器。在 X Window 系统，这个过程中浏览器被称为客户，因为是浏览器发出显示的请求，而把系统中负责显示的系统功能称为 X 服务器。

1984 年 Bob Scheifler 和 Jim Gettys 制定了 X 的设计原则。几十年以来，最新的 X 仍然能与最初的版本兼容，这些原则居功至伟。这些原则中第一条原来的内容是“X 原则上不增加新的功能，除非缺少该功能就不能建立一个完整的应用程序”，后来修订为现在的内容，其余内容 24 年未曾有过变动。这些原则如下。

- X 原则上不增加新的功能，除非存在迫切需要新功能的应用程序。
- 提供系统的灵活扩展性，而不是提供复杂的实现细节。
- 如果没有真正理清问题的来龙去脉，就先别想着去解决问题。
- 如果一个问题的解决方案看上去太复杂，就应该改用简单的方法解决。
- 提供实现的机制而不是实现的细节。

X Window 系统是一种设备无关的图形界面。只要遵守其 Xprotocol 协议，所有软件不经重新编译即可在 X Window 系统中正常使用，并且显示图形界面。任何硬件只要遵守 X Protocol，就可以进行相应的窗口显示工作。所以 X Window 系统的应用范围涵盖了从高端的商务、科研大型机到个人计算机的各个层次。

X 只定义了图形界面的基本框架，如显示窗口内容及移动窗口，以及处理鼠标键盘输入等。至于人机交互的其他界面并没有涉及，这些工作由基于 X 的另一些程序（如窗口管理器等）来实现。这样，就有了许多种实现方式。表现形式是：许多使用了同样的 X 的系统看起来有极大的差异。如 GNOME 采用的窗口管理器是 metacity，而 fvwm 使用的窗口管理器是 fvwm。

20 世纪末，采用的 X Window 实现一般是 XFree86。但从 2004 年起一般的发行版均使用了 XFree86 的一个分支 X.Org。X.Org 采用的 X 是 1987 年 9 月发布的 X11，而现在 X11 已经发布到 X11R 7.1 了。

X 的适用范围很广，在 UNIX、类 UNIX、Mac OS 中均得到了广泛的应用。只有 Windows 不怎么支持 X。不过在 Windows 中，用第三方软件是可以支持 X Window 的，如 Cygwin。X

虽然在 UNIX、类 UNIX 系统中取得了很大的成功，击败了许多竞争对手，但其实现也存在着很多的缺点。

- X 对于某些硬件的支持度比较差。这是由于某些高端硬件为了保持商业机密，对于硬件驱动信息采用了保密措施，使得无法开发基于该硬件的自由驱动程序，所以对于某些硬件而言，X 往往无法提供最高性能的支持。
- X 是硬件设备与图形界面的一个中间层，存在中间层从某种意义上来说占用了系统资源，增大了系统开销。现在 X 正努力实现直接读取硬件设备来改善性能。
- X 的稳定性也存在着潜在的问题，有时甚至会导致系统的崩溃。
- X 没有通用的声音子系统，出现了许多互不兼容的声音子系统。
- X 的显示与打印是两个独立的子系统，不像微软 Windows 系统显示和打印只是同一个设备的两种表现形式。X 对于打印功能支持得并不好。

3.6.2 发展历史

X 发源在 MIT（麻省理工学院）这个科学奇才和科研成果层出不穷的圣地，最早发布在 1984 年。由于其具有良好的扩展性与性能，逐渐为 UNIX、类 UNIX 系统（如 Linux）等系统采用，成为其显示系统的基础。X 开发的初衷是开发一套可以在 UNIX 系统中通用的图形界面，由于其是在一个叫 W 的实验性窗口系统基础上开发出来的，所以按字母表中 W 的下一个 X 来命令。

1989 年，X 的重要分支 XFree86 建立。1994 年，一个里程碑式的版本 X11R6 发布，这是 X Window 协议最成熟稳定的版本，该版本确定了 X Window 的基本特性，这些特性到现在也没有变动过。2004 年 9 月发布了 X11R6.8，这是一款现代化的版本，支持很多现代化的特性，如透明桌面等复杂特性。在 X.Org 基金会和 freedesktop.org 的共同发力下，现在的 X 逐步在实现一个真正的图形界面，而且试图解决其存在的问题，如直接读取硬件等。

3.6.3 字体的安装概述

系统的默认字体在 Ubuntu8.04 里已经有了很大的改进，但追求完美之心人皆有之，许多人总想用更加美观的字体来代替系统的默认方案。不过，好在这个工作也不难。在设置系统字体之前，先对 X Window 的字体安装进行简单的介绍。

X Window 包括了两套字体系统：核心字体系统和 XFT 字体系统。核心字体系统历史悠久，传统的 X 程序都是使用核心字体系统。但是随着计算机系统硬件的不断进步，对于字体的显示要求越来越高，核心字体系统对于各种复杂系统和高要求的字体显示要求越来越力不从心了。为了更好地支持矢量字体，实现抗锯齿、子像素光栅化等高级特性，XFT 字体系统逐渐在许多场合取代了核心字体系统。

但是核心字体系统并没有完全退出历史舞台，很多老的 X 程序仍然采用核心字体系统，要把这些程序全部迁移到 XFT 字体系统还要做许多工作。因为核心字体系统有占用系统资源少、处理相对容易等特点，在控制台下工作的许多程序仍然只支持核心字体系统。

所以，这两种字体系统在可以预见的未来还会暂时共存下去。在安装字体的时候，就需要同时考虑到这两种字体的不同特点分别处理。

3.6.4　安装核心字体

安装核心字体相对来说，需要的步骤稍微多些。不过，也不是很难，跟着下面的介绍做就可以了。安装步骤如下。

（1）建立一个目录，用来存入这些核心字体。本例中建立的目录名为 myfonts（在控制台中执行如下命令，下同），使用的命令是 mkdir myfonts。

（2）更改工作目录到新建的目录中。使用命令 cd myfonts。

（3）把字体复制到目录。使用命令“cp /media/winXP/fonts/simsun.ttf ～/myfonts”。

（4）在控制台中该目录下执行 sudo chmod go+rx *.ttf 命令，改变字体的权限。

（5）生成 fonts.scale 与 fonts.dir。执行下列两条命令即可。

```
mkfontscale
#这条命令是用来生成 fonts.scale 文件
mkfontdir
#这条命令是用来生成 fonts.dir 文件，注意这两条命令的顺序不能颠倒，否则生成的 fonts.dir 是空文件。其实这两个文件的内容是一样的，因此也可以直接生成“cp fonts.scale fonts.dir”
```

（6）在控制台中运行“sudo gvim fonts.scale”，意思是用 root 用户身份打开该文件。

（7）在文件中加入这行“simsun.ttf -misc - simsun - medium - r - normal --0-0-0-0-p-0-gbk-0”。注意关键是最后的 gbk-0 这几个字符。

（8）该文件第一行的数字代表了字体的个数，现在加入了一个字体，就要给字体计数器的值加 1。也就是说，应该将第一行的数字修改为其值加 1 后的数字，如文件中第一行的数字为 25，就应将其修改为 26。



第 5 步使用 mkfontscale 与 mkfontdir 命令生成的 fonts.scale 文件中缺少对 gbk 字体的支持，需要手工编辑 fonts.scale，在其中加入相关的内容。第 6 步至第 8 步均是编辑 fonts.scale 文件。

（9）fonts.dir 文件也要参照 fonts.scale 进行相应的修改，使这两个文件保持一致。如果不进行这一步，就无法使用 gbk 字体。

（10）使用命令“sudo gvim /etc/X11/xorg.conf”打开文件“/etc/X11/xorg.conf”进行修改。在该文件中找到 Section“Module”，添加 load“freetype”，意思是加载 freetype 模块。修改后的结果可能像这样：

```
Section "Module"
…
Load "freetype"
…
EndSection
```

再在该文件中找到 Section“Files”区域，在其中添加 FontPath 后面跟安装字体的目录，修改完该文件后，文件内容可能像这样：

```
Section "Files"
…
 FontPath "/home/westnorth/myfonts"
```

```
…
EndSection。
```

（11）现在已经完成了对 X 核心字体的配置工作。大多数不太旧的版本的 Linux 图形程序都支持 fontconfig。为了让 fontconfig 使用这个点阵字体，需要执行命令：

```
fc-cache -f -v
```

（12）按下 Ctrl+Alt+Backspace 组合键，重启 X，让字体更改生效。

如果重启 X 后，用“xlsfonts”命令还是看不到刚安装的新字体，再重启一下计算机应该就可以看到新安装的核心字体。



一般而言，用 xlsfonts 命令显示的字体极多，不容易分辨哪些是中文字体。可以使用命令 “xlsfonts | grep gb2312”或者“xlsfonts | grep gbk” 来查看相关中文字体，前者查看 gb2312 中文字体，后者查看 gbk 中文字体。

3.6.5 安装 XFT 字体

安装 XFT 字体相对容易得多，只需以下几个步骤即可。

（1）在主目录建立一个名为.fonts（注意在 f 前面有一个“.”）的隐藏目录。

（2）把要安装的字体移动到该目录中。

（3）打开终端，执行如下命令，更改当前目录到字体所在的目录：

```
cd .fonts
```

（4）创建两个文件，分别控制字体的缩放与一些特效，如粗体、斜体等。使用命令“mkfontscale；mkfontdir”。

（5）刷新字体缓存，可以使用命令 fc-cache -fv 来实现。



这种方法安装的字体不是核心字体，所以许多只能使用核心字体的程序（特别是一些字符界面的程序）无法使用。

3.7 小结

本章介绍了 Linux 系统用户界面中最重要的内容桌面环境，包括桌面环境的基本介绍，以及美化系统的方法。桌面系统是用户第一次接触 Linux 时首先使用的功能，所以本章对初次接触 Linux 系统的用户来说是很重要的入门内容。本章的难点是安装系统核心字体的内容，而且如果没有正确安装字体，有些字符界面程序无法正常运行。而这些字符界面程序在系统管理中往往具有很重要的意义，可见掌握核心字体的安装方法具有十分重要的实践价值。

第 4 章　Linux 基本配置

对于 Linux 使用者来说，能打开一个应用程序，进行简单的系统外观设置设置是远远不够的，用户还应该了解系统的硬件信息。简单地优化硬件性能，以方便管理硬件。同时还应该知道启动过程，能让系统加载自己想要的程序。作为网络时代的用户，让 Ubuntu 系统连入互联网也是必不可少的技能。中文编码有其特殊性，为此用户还应了解 Locale 的概念及设置。本章介绍这些系统基本配置知识。

4.1　硬件管理

要想让系统中的硬件完美地工作，就需要对其进行适当的管理，而管理的前提是先知道硬件信息，才能“知人而善用”。虽然现在 Linux 各发行版对硬件支持度一般都比较理想，但有时可能还会需要对其进行管理。本节将介绍查看硬件信息的相关命令，以及系统管理硬件相关信息的重要目录“/proc”。

4.1.1　硬件管理概论

用户对于 Linux 的很重要的一个印象是其对硬件的支持很不好。很多硬件在 Windows 下面使用正常，但是在 Linux 中却因为没有驱动程序而无法使用。可以说这种情况还是几年前的往事，现在的发行版对硬件的支持度极好。普通的硬件根本不需要任何驱动，如显卡、声卡、扫描仪等均可直接在 Linux 中使用。对于现在的发行版，硬件管理的意义一般都不再是给那些 Linux 不识别的硬件安装驱动程序。

但是有时候，为了更好地发挥硬件的性能，可能还需要给其安装驱动程序。这时，获得硬件信息就是很重要的事情。但获得硬件信息并不是说看看发票说“哦，这是一块创新的声卡”这么简单。一般硬件最主要的部件是核心芯片，而核心芯片的提供厂商是很有限的。例如，CPU 只有 Intel 和 AMD 两个，显卡则主要是 ATI、NVIDIA、SIS 等。一般显卡只是制造厂商购买了显卡芯片，然后放在一块精心设计的电路板上以实现其功能。其他板卡如网卡，主板均是同样的道理。

明白了这些，也就很容易理解 Linux 如何对硬件提供支持了。Linux 只是简单地读取硬件的芯片信息，然后与自己库里的驱动程序进行比较。发现有匹配的程序，就用该驱动程序加载硬件，实现各种功能。Linux 提供了多种查询硬件信息的命令，可以轻易地查看硬件相关信息，以帮助用户使用最匹配的驱动程序和获得适当的帮助。下面简单地介绍一下各种查看硬件及系统信息的命令。

4.1.2　列出所有 PCI 设备

列出所有 PCI 设备可以使用 lspci。这是 Linux 硬件信息查询最常用的命令。这个命令可以显示系统中所有 PCI 设备。由于系统中大多数设置均是 PCI 设备（网卡、声卡、1394 火线接

口、集成显卡等均是 PCI 设备），所以该命令的重要性也就不言而喻了。lspci 是读取 pciids 数据库，如果出现硬件信息无法正确识别，则可以使用“sudo update-pciids”来更新一下 PCI 的数据库，然后 Ubuntu 就可以正确识别 PCI 设备了。lspci 的参数很有意思，重复有的参数意味着程度的递增。下面是一些参数及意义。

- v：查看硬件的详细信息，如 IRQ 中断等。如果网卡不能正常使用，那么可以用这个命令查看输出信息，查看是否 IRQ 存在冲突。如果硬件出现无法驱动的情况要寻求帮助，就应该把 lspci -v 命令输出的相应硬件信息全部列出来，这样别人才能根据这些信息提供可信的帮助。
- vv：获得比 v 参数更详细的信息。这个就显示了基本全部可用的信息。
- vvv：获得比 vv 更详细的信息。
- n：显示供应商和设备代码，就不必在列表中慢慢地查找了。
- nn：同时显示供应商、设备代码及名字。
- x：显示硬件基本设置信息的二进制数据。
- xxx：显示完整的硬件信息的二进制数据。
- xxxx：显示扩展硬件信息（如 PCI-X2.0 或 PCI-E）的二进制数据。

如果硬件有了问题，可以使用 lspci -vvx 或 lspci -vvxxx 来获得硬件信息。然后将其公布，硬件厂商或热心的网友就会根据这些信息提供帮助。

4.1.3 查看内核信息

由第 1 章的相关介绍可以知道，Linux 内核是 Linux 操作系统的核心所在。由于不同版本的内核有不同的特性，为了内核升级或安装特定程序的需求，可能需要知道如内核名称、当前操作系统名称等信息，这时就可以使用 uname 命令。其使用格式如下：

```
uname [参数]
```

uname 可以用来查看内核版本、内核名称、计算机的体系结构等信息。这些均可以通过参数获得，这些参数如下所示。

- -a、--all：显示所有的系统信息。
- -s、--kernel-name：显示内核名称，如显示可能类似于 Linux。
- -n、--nodename：显示网络节点名，就是网络上别人看到本机的名字。
- -r、--kernel-release：显示内核版本，如显示可能类似于 2.6.24-16-generic。
- -m、--machine：显示当前计算机的结构，如显示可能类似于 i686。
- -o、--operating-system：显示当前操作系统名称，如显示可能类似于 GNU/Linux。



如上所述用顿号隔开的参数“-r、--kernel-release”意思是在使用时可以选择任意一种方式如“-r”或“--kernel-release”即可。

```
$ uname -a              #显示所有的系统信息，与使用“uname  --all”功能相同
Linux mylaptop 2.6.27-10-generic #1 SMP Fri Nov 21 12:00:22 UTC 2008 i686
GNU/Linux
```

输出依次显示了内核名称（Linux），网络节点名（mylaptop），内核版本（2.6.27-10-generic），

系统结构（i686），当前操作系统名称（GNU/Linux）。

4.1.4　显示 Ubuntu 版本信息

如果远程连接到一个计算机，为了方便对其进行操作，可能需要知道其采用的是哪一种操作系统，以及其版本信息等，这样可以给操作带来许多方便。如果要知道当前操作系统的版本信息，就可以使用 lsb_release，其使用格式如下：

```
lsb_release -a
```

这条命令用来显示 Ubuntu 的版本信息。在系统中有一个文件的内容也保存着相同的信息，这个文件是“/etc/issue”。该命令的输出如下：

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu                  #发行版名称：Ubuntu
Description:    Ubuntu 8.10             #发行版的描述信息
Release:    8.10                        #发行版本
Codename:   intrepid                    #发行版代号
```

4.1.5　查看内核加载的模块

Linux 系统中，可以将部分程序，如一些硬件的驱动程序作为模块加载到系统。要查看系统加载的所有模块，就可以使用 lsmod。该命令的输出应该类似如下所示：

```
$lsmod
Module                  Size  Used by
i915                   38144  2
drm                    86056  3 i915
binfmt_misc            16904  1
ppdev                  15620  0
acpi_cpufreq           15500  0
cpufreq_ondemand       14988  2
cpufreq_conservative    14600  0
cpufreq_powersave        9856  0
cpufreq_userspace      11396  0
cpufreq_stats          13188  0
freq_table             12672  3 acpi_cpufreq,cpufreq_ondemand,cpufreq_stats
```

输出共分为 3 列，第一列是模块名，第二列是该模块的大小，第三列表示了该模块被哪一个进程所使用。如“cpufreq_stats 13188 0”意思是 cpufreq_stats 模块的大小是 13 188 字节，被进程 0 使用。

4.1.6　查看 USB 设备

随着现代 USB 设备的增多，在很多时候需要查看 USB 端口的使用情况，来判断 USB 端口的使用情况，以及获得 USB 设备的一些技术参数。这时就可以使用 lsusb 命令。lsusb 是 Linux 专门

用来显示 USB 设备的命令，可以获得丰富的 USB 设备的信息。其具体用法可参见如下两个示例：

```
$lsusb
Bus 005 Device 001: ID 0000:0000    #如果该端口未被使用，则显示为 0000:0000
Bus 001 Device 003: ID 15d9:0a37    #如果端口被 USB 设备使用，则输出可能是这样
```

lsusb 的参数 v 比较有用，可以输出详细的 USB 信息，包括 USB 速度等，其输出内容如下所示：

```
$ lsusb -v
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
#第 5 总线的第一个设备，ID 为 1d6b:0002，接口类型是 USB 2.0
Device Descriptor:      #下面是设备的一些具体信息的描述
  bLength              18
  bDescriptorType       1
  bcdUSB              2.00
  bDeviceClass          9 Hub
  bDeviceSubClass       0 Unused
  bDeviceProtocol       0 Full speed (or root) hub
  bMaxPacketSize0      64
  idVendor          0x1d6b Linux Foundation
  idProduct         0x0002 2.0 root hub
.................
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x81  EP 1 IN
        bmAttributes            3
          Transfer Type            Interrupt
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0004  1x 4 bytes
        bInterval              12
```

输出的结果显示了系统有一个 USB 设备，其接口标准是 2.0，而且目前没有使用。

4.1.7 查看网卡状态

有时候，如果网络连接出现故障，在使用其他方式排除故障前，应该先行检查网卡是否被正确识别。这是由于网卡是网络连接的主要设备，如果网卡没有被系统正确识别，则网络是不可能正常连接的。Linux 查看网卡状态可以使用 ethtool 命令，其具体用法如下例所示：

```
$ sudo ethtool eth0                 #查看 eth0（第一个网卡）的状态信息
Settings for eth0:
Supported ports: [ TP MII ]
Supported link modes:   10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
Supports auto-negotiation: Yes
Advertised link modes:  10baseT/Half 10baseT/Full
```

```
                                100baseT/Half 100baseT/Full
    Advertised auto-negotiation: Yes
    Speed: 10Mb/s
```

由输出信息可以看出，其是一个 10/100Mbps 自适应网卡。这些信息在查询网卡信息时比较有用。

4.1.8　查看 CPU 信息

有一个特别的目录，就是“/proc”。该目录里面存放着各种系统信息，每一个文件对应不同的信息，当然其中包括显示 CPU 信息文件。存放着 CPU 信息的文件是/proc/cpuinfo，可以用如下命令来查看：

```
cat /proc/cpuinfo
```

其用法及输出如下：

```
$ cat /proc/cpuinfo
processor   : 0                     #CPU 编号
vendor_id   : GenuineIntel          #CPU 架构
cpu family  : 6                     #CPU 体系编号
model       : 14                    #CPU 的型号
model name  : Genuine Intel(R) CPU          T2060  @ 1.60GHz
#CPU 的名称为 Genuine Intel(R) CPU T2060，其主频为 1.60GHz
stepping    : 12                    #CPU 的倍频
cpu MHz     : 800.000               #CPU 的实际速度
cache size  : 1024 KB               #CPU 缓存大小
physical id : 0                     #CPU 的物理编号，在多个 CPU 时很有用
siblings    : 2
core id     : 0
cpu cores   : 2                     #CPU 核心数量为 2，意思是双核的 CPU
```

由输出可以看出，该 CPU 是 Intel 的双核处理器，T2060，频率为 800MHz，一级缓存 1024KB。其他各种信息在上文已经在注释中说明，可自动查看一下。

4.1.9　显示当前硬件信息

sudo lshw 可以显示完整的系统硬件信息，包括内存、CPU 版本和速度缓存、系统总线速度等。得到输出可能需要等待一段时间，但提供的完整信息会让等待物有所值。还可以这样使用。

- lshw -short：用短格式列出系统信息。
- lshw -class disk -class storage：列出所有的磁盘和存储控制设备。

由于输出内容较多，而且输出内容比较繁杂，这里就不列出输出结果了，有兴趣的读者可以自行查看命令的输出结果，并分析输出的意义。

4.1.10　查看当前的内存使用情况

内存的使用情况直接关系到系统的性能，一个有经验的用户会经常查看其内存使用情况，

以判断当前系统的健康状况。Linux 的查看内存的命令是 free。其用法如下（以#开头的注释行为笔者所加）：

```
$free -mt
             total       used       free     shared    buffers     cached
Mem:          2017        931       1086          0         57        532
#使用的内存的数量，总共有 2017MB，使用了 931MB，空闲 1086MB
-/+ buffers/cache:        340       1676
#实际使用的物理内存数量为 340MB，空闲了 1676MB
Swap:          486          0        486
#交换分区总数为 486MB，使用了 0MB
Total:        2504        931       1572
#总共有内存 2504MB,使用了 931MB，空闲 1572MB
```

Mem 行输出内存名义使用量，可以看到已经使用了 931MB。而-/+ buffers/cache 行是内存真正的使用量，可以看到实际使用了 340MB。free 命令的参数如下。

- b、k、m、t：分别以字节、千字节、兆字节、吉字节为单位显示内存使用情况。
- t：显示内存和虚拟内存的使用总量。
- s n：间隔 n 秒动态显示内存使用情况。如 free -s 10，每隔 10 秒显示一次内存使用情况。
- c n：统计前 n 秒内存使用情况。

4.1.11 显示内核输出信息

dmesg 显示的内容很丰富，从开机自检的信息到系统运行时检测到的硬件信息均可包含。所以，如果硬件在使用中有什么问题，可以通过 dmesg 查看相关信息来判别错误的原因。

```
earthfans@mylaptop:~$ dmesg
…
[  655.805128] usb 5-5: new high speed USB device using ehci_hcd and address 2
[  655.938218] usb 5-5: configuration #1 chosen from 1 choice
[  656.056079] usb 5-5: reset high speed USB device using ehci_hcd and address 2
[  657.027003] ndiswrapper: driver zd1211bu (Wireless,06/25/2007,6.22.0.0)
loaded
[  657.789834] wlan0: ethernet device 00:21:27:b8:36:05 using NDIS driver:
zd1211bu, version: 0x6160000, NDIS version: 0x501, vendor: '802.11 b+g Wireless
LAN', 0ACE:1215.F.conf
[  657.820985] wlan0: encryption modes supported: WEP; TKIP with WPA, WPA2,
WPA2PSK; AES/CCMP with WPA, WPA2, WPA2PSK
[ 1036.205204] CE: hpet increasing min_delta_ns to 15000 nsec
…
```

上文是 dmesg 的一部分，显示了一个 USB 无线网卡插入后系统的加载过程。dmesg 有一个重要的参数 c，是用来清理缓冲区的内容，但由于 dmesg 显示的信息是动态生成的，所以在下次开机后，仍然可用 dmesg 来查看新的信息。

```
dmesg -c
```

4.1.12　显示当前系统中 HAL 硬件信息

HAL（Hardware Abstraction Layer）翻译过来是硬件抽象层，这是 GNOME 的基础库，提供了对于硬件的管理功能。lshal 显示系统中 HAL 数据库中的内容，显示的内容与 dmesg 有些类似，不过更为翔实。lshal 还可以使用参数 t 来显示树形列表，如“ls -t”命令的显示类似这样：

```
pci_8086_27d2
  pci_8086_27d0
  pci_8086_27d8
    pci_8086_27d8_sound_card_0
      pci_8086_27d8_sound_card_0_alsa_capture_2
      pci_8086_27d8_sound_card_0_alsa_playback_1
pci_8086_27d8_sound_card_0_alsa_capture_1
```

输出显示了一个声卡设备，包括捕获、回放功能等。

4.1.13　相关硬件信息的目录及配置文件

下面是一些硬件的配置文件所在地，以及一些硬件的配置工具，可以在使用时查阅。

- 显示器和显示卡的配置文件是/etc/X11/xorg.conf。
- 声卡的配置文件是/etc/modprobe.conf 或 /etc/modules.conf。
- 文件系统的配置文件是/etc/fstab。
- 声卡的终端配置工具是 alsamixer。
- 内核模块支持的硬件驱动位于/lib/modules/内核版本/kernel/drivers 目录中，应为/lib/modules/2.6.24-16-generic/kernel/drivers。要获得内核版本的信息，可以使本节介绍的命令 uname。

4.2　硬件信息目录/proc

/proc 看起来似乎是一个普通的系统目录，但其功能强大，而且实现也很特殊。其全称是“process information pseudo-filesystem”，译成中文大概的意思是“进程信息伪装文件系统”。/proc 是一个核心数据结构的接口，里面的内容是核心的一些信息（如硬件信息、内存信息，系统运行进程的信息）。对用户来说该目录中绝大多数内容是只读的，但内容还是随着系统状态不同而变化的。

这个目录从某种意义上来说其实并不存在，因为其没有挂载在任何硬盘上，只是将自身的信息注册到了 VFS（虚拟文件系统）中，只有对其进行操作，才会动态产生信息。但从另一个意义来看，这个目录又是存在的，因为可以在命令行或文件管理器中进入这个目录，并查看文件的内容。

4.2.1　查看该目录下的文件

在命令行运行命令“cd /proc”，进入该目录。然后使用命令 ls 查看其中的文件目录信息，显示如下所示：

```
1     2336  4967  5331  6868  7580        interrupts    self
```

```
10    2537  4970  5434  6870  7636        iomem        slabinfo
11    2739  4992  6     6876  8           ioports      stat
136   3     5     6734  6884  9           irq          swaps
1516  3014  5009  6739  6885  acpi        kallsyms     sys
1517  3027  5023  6763  6888  asound      kcore        sysrq-trigger
1550  4     5036  6764  6894  buddyinfo   key-users    sysvipc
1552  4264  5099  6815  6897  bus         kmsg         timer_list
```

可以看到其中有许多以数字命名的目录。每个数字目录代表一个当前运行的进程，数字就是进程号。除了这些数字目录外，还有一些系统信息文件。这些文件每个代表一个重要的系统信息。下面将分别予以介绍。

4.2.2 系统信息文件

/proc 目录中的这些系统信息文件代表了重要的意义，而这些文件其实也是在动态变化的。正如前面介绍的，这些文件，只有在用户查看时才会被动态创建。下面列出部分文件的意义。

- /proc/cpuinfo：CPU 的信息（频率、缓存大小等）。
- /proc/meminfo：物理内存、交换空间等的信息。
- /proc/mounts：已加载的文件系统的列表。
- /proc/devices：可用设备的列表。
- /proc/filesystems：被支持的文件系统。
- /proc/modules：已加载的模块。
- /proc/version：内核版本。
- /proc/cmdline：系统启动时输入的内核命令行参数。
- /proc/dma ：显示当前使用的 DMA 通道。
- /proc/interrupts：显示使用的中断。
- /proc/ioports：当前使用的 I/O 端口。
- /proc/kmsg ：核心输出的消息。
- /proc/ksyms ：核心符号表。
- /proc/loadavg：系统平均负载情况。
- /proc/net ：网络协议状态信息。
- /proc/uptime ：系统启动的时间长度。
- /proc/version ：内核版本信息。

4.2.3 进程相关目录

进程相关目录中包含了许多数字目录，每一个数字目录对应于一个进程。同时每个数字目录中包含了进程号为该数字的进程相关的信息。下面以一个进程号为 7317 的进程为例介绍这些数字目录中的内容。

```
$ps                    #列出当前进程
#输出如下所示
  PID TTY          TIME CMD
 7317 pts/1    00:00:00 bash
```

```
 7370 pts/1    00:00:00 ps
#可以看到进程号为 7317 的进程是 bash。下一步就应进入 bash 进程的相应目录 7317
$cd /proc/7317       #更改当前目录到 7317 子目录
$ls                  #显示文件列表
#输出如下所示，正文将解释这些文件的意义
attr     cmdline      environ fdinfo loginuid mounts    oom_score smaps status
auxv     coredump_filter exe     io     maps     mountstats root     stat  task
clear_refs cwd          fd     limits mem     oom_adj    sched     statm wchan
```


可以看到，这些目录里面又有许多的文件。这些文件同样包含着重要的信息，而且大部分是纯文本文件，可以用任意的文本编辑器打开查看其内容。其中部分文件的意义如下所示。

- cwd：当前工作目录，是符号连接文件。连接的源文件是进程的目录/proc/7317。
- exe：可执行文件，也是符号连接文件。连接的源文件是当前进程名 bash。
- environ：里面包含所有的环境变量。
- mounts：显示了加载的文件系统的信息。
- limits：显示了系统的各种最大限制数量（如最大打开的文件数目、最大进程数、最大堆栈数等）。
- status：登录用户的相关信息。

4.3　启动过程详解

系统启动时，会读取一些目录里的重要脚本。Linux 的配置文件为了方便人的阅读，均是以纯文本的形式存放。所以，通过阅读、分析系统启动脚本，可以深刻理解系统启动过程。用户还可以定制启动脚本，对系统的深入控制提供了极大的灵活度。这也是 Linux 的重要特色。本节主要分析 Linux 的启动过程，以及介绍如何管理自启动项目。

4.3.1　Linux 启动过程

系统启动过程其实并不神秘，而且系统启动的原理和 20 多年前没有什么大的差别。系统加电之后，CMOS 负责将启动盘第一扇区的内容装入地址为 0x7c00 的内存中，然后跳转到 0x7c00 处开始执行。这时 Linux 的 bootsect.S 就登台了，其将系统核心代码循环移动到内存中。这些代码运行后，Linux 的核心就运行起来了。

Linux 启动程序首先加载核心代码负责数据区、idt 表、页表和寄存器的初始化，同时进行必要的状态检查。然后运行 start__kernel 进程，该进程继续初始化（这里初始化的内容就更多了，如获取硬件的驱动信息，设置各种入口地址，具体就不一一列举了，有兴趣的读者可以下载 Linux 的源码，在“/init/main.c”代码中查看“start_kernel”函数，里面有所有加载的内容，而且还有相关的注释，应该很容易看懂。）。

从这里开始，内核就能够挂装各文件系统。内核挂装了根文件系统，并已初始化所有的设备驱动程序和数据结构等之后，就通过启动一个叫 init 的用户级程序，完成引导进程。init 进程是 Linux 的根进程，所有的进程都是它的子进程。

一般发行版均采用 inittab 通过改变运行级别来启动或停止服务，但 Ubuntu 采用了 upstart

来处理。inittab 有文件分级的概念，而使用 Upstart init daemon 的 Ubuntu 系统没有运行级别的概念。为了将基于运行级别的系统平滑移植到基于事件的系统，并为面向其他发布版的软件提供一定的兼容性，Ubuntu 使用 Upstart 模拟了运行级别。对于运行级别，应该有所了解。不同的运行级定义如下如示。

- 0：停机，处于该状态的程序永远不会被运行。
- 1：单用户模式。
- 2：多用户，没有 NFS。
- 3：完全多用户模式（标准的运行级）。
- 4：尚未定义。
- 5：X11（X Window）的运行级。
- 6：重新启动，处在该状态的程序一直在重新启动。

下面再介绍 init 如何运行脚本来启动所需的服务，在知道 init 如何运行之前，还应该知道有些预备知识。

1．/etc/event.d/目录

/etc/event.d/目录是 upstart 的核心，upstart 将所有进程的触发、停止等都看做 event（事件）。/etc/event.d/中就存放了目前 upstart 需要识别的 event。其中主要有三种 rc-default，rcX（x=0，1，...6，S）以及 ttyX。这个 rc-default 就类似于其他发行版的 inittab 文件，用来设置默认运行级别。使用命令“vi /etc/event.d/rc-default”查看其源文档，可以看到其为了与其他发行版兼容，还包含着从 inittab 里获取运行级别的内容。但对于 Ubuntu 来说，最重要的是以下几行：

```
20 else
21        telinit 2
22    fi
23 end script
```

要更改默认运行级别为三级，只需将第 21 行“telinit 2”修改为“telinit 3”。修改为其他级别也可进行类似的操作。

2．/etc/init.d

/etc/init.d 是一个目录，里面的内容是系统启动所需的全部脚本，以及系统中安装的服务的启动脚本。

3．/etc/rcS.d

/etc/rcS.d 是默认运行的一个脚本，系统启动后会先运行该目录中的脚本。里面的内容用 ls -l 查看类似于：

```
    lrwxrwxrwx 1 root root  24 2008-10-11 03:44 S01mountkernfs.sh -> ../init.d/
mountkernfs.sh
    lrwxrwxrwx 1 root root  19 2008-10-11 03:58 S01readahead -> ../init.d/
readahead
    lrwxrwxrwx 1 root root  21 2008-10-11 03:44 S02hostname.sh -> ../init.d/
hostname.sh
    lrwxrwxrwx 1 root root  24 2008-10-11 03:44 S06keyboard-setup -> ../init.d/
keyboard-setup
    lrwxrwxrwx 1 root root  41 2008-10-11 03:45 S07linux-restricted-modules-
common -> ../init.d/linux
```

```
...
lrwxrwxrwx 1 root root  20 2008-10-11 03:49 S70x11-common -> ../init.d/x11-
common
lrwxrwxrwx 1 root root  21 2008-10-11 03:44 S80bootmisc.sh -> ../init.d/
bootmisc.sh
lrwxrwxrwx 1 root root  17 2008-10-11 03:44 K85urandom -> ../init.d/urandom
lrwxrwxrwx 1 root root  27 2008-10-11 03:44 S90console-screen.sh -> ../init.
d/console-screen.sh
```


可以看到，里面的脚本其实都是/etc/init.d 中脚本的一个符号连接。但是连接过来后，给前面加了一个字母和两个数字。如将 mountkernfs.sh 变为 S01mountkernfs.sh。这样做有两个原因，这两条原因在其他目录里，如/etc/rc2.d、/etc/rc3.d 里也是适用的。

- 第一个字母 S 是告诉系统该脚本处于运行状态，系统启动时会运行所有以 S 开头的脚本。
- 第二、三位的数字是运行的优先级，数字越小，则越先运行。如在上面列出的文件中，系统启动时最先运行 S01mountkernfs.sh，最后运行 S90 console-screen.sh（因为这两个脚本的第一位字母均是 S，S 代表运行）。S01mountkernfs.sh 的数字是 01，要大于 S90 console-screen.sh 的数字 90，可以先运行。不运行 K85urandom，因为其第一位的字母是 K，代表启动时不运行。

4.3.2　init 进程在启动时所做的工作

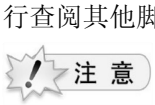
前面介绍了系统启动过程，也简单介绍了 init 的一些功能。可以知道 init 是系统启动时一个极其重要的程序。本节将具体介绍 init 进程的所做的具体工作。

1．运行系统级别的自启动脚本

/etc/rcS.d 目录存放的系统级的自启动服务的脚本，init 会首先运行该目录中名字以 S 开始的脚本。这里面的脚本完成的工作是系统初始化及设置。以一个 rcS.d 目录中优先最高的脚本 S01mountkernfs.sh 为例来看其进行了哪些操作，脚本的部分内容如下：

```
19    do_start (){
20    # Mount proc filesystem on /proc
21    domount proc "" /proc -onodev,noexec,nosuid
22    # Mount sysfs on /sys
23    domount sysfs "" /sys -onodev,noexec,nosuid
24    # Mount /var/run and /var/lock as tmpfs.
25    domount tmpfs "" /var/run -omode=0755,nodev,noexec,nosuid
26    domount tmpfs "" /var/lock -omode=1777,nodev,noexec,nosuid
```

可以看到第 21 行加载了重要的文件系统/proc，第 23 行加载/sys，然后是几个其他目录的挂载。从这些脚本里可以看到在完成启动过程前系统是做了很多工作的。如果有兴趣，可以自行查阅其他脚本。



除非具有专家级的技能，否则不要轻易更改系统启动脚本。

2．确认运行级别

打开/etc/event.d/rc-default 文件，确定默认的运行级别（Ubuntu 的默认级别是 2）。当然，

从前面的介绍中可以知道，运行级别只是 Ubuntu 为了与其他系统兼容而保留的，可以说实际意义并不太大。

3．运行相应级别的 rcX 文件

得到运行级别后，触发相应的 runlevel 事件，开始运行/etc/event.d/rc2。rc2 中第 19 行很重要，这行的功能是运行 rc，运行参数是 2。内容如下：

```
exec /etc/init.d/rc 2
```

4．运行 rc 程序

以参数 2 运行/etc/init.d/rc 程序。该程序的功能是进行一些基本设置，然后运行/etc/rc2.d 中的脚本。

5．运行用户级的启动脚本

运行/etc/rc2.d/中的脚本。/etc/rcX.d/中的脚本按事先设定的优先级依次启动，完成了启动过程，进入登录环节。

4.3.3 管理自启动项目

有的程序使用频率很高，基本上每次开机后的第一件事就是打开该程序，这时一种效率比较高的方式就是让其在系统启动时自动运行。这种方式用过 Winodws 的用户都很熟悉，而且可能还有一些不太愉快的回忆，安装了一些软件后，如 MSN、QQ 之类，系统启动后这些程序就会自动启动。而且动辄系统会莫明其妙地自动启动一些不怎么想要的程序，使得系统缓慢、不稳定，还可能会泄露自己的隐私信息。

但 Linux 不会这样，Linux 对于自启动软件管理很严格，只有超级用户才有权限更改，而且启动方式很透明，用户随时都可以检查自启动程序，轻易令其不再随机启动。从 4.3.2 节可以看到，Linux 中不仅可以自定义启动普通程序，对于系统启动过程和底层设置也是可以更改的。

有许多方法可以在 Linux 启动时自动启动服务或指定的程序。初级用户可以选择使用图形界面进行管理，而高级用户可以通过命令行方式修改自动启动的程序，甚至可以自己编写相应的脚本来对程序进行相应的初始化，然后进行启动。下面是这些方法的介绍。

1．图形界面管理自动加载程序

对于初学者而言，使用图形界面无疑是最为方便的。Linux 提供了图形界面管理的方法，选择主菜单“系统”|“首选项”|“会话”命令，打开“会话首选项”对话框，如图 4.1 所示。

在“额外的启动程序”列表框中列出的是系统启动时自动加载的程序，前面有对号的为自动加载的程序。如果要使该程序在系统启动时不自动加载，则取消该程序前的对号。如图 4.1 所示，Tracker 不自动加载，而 Network Manage 自动加载。如果要让列表中不存在的应用程序自动加载，则需要单击“添加”按钮。接着会弹出对话框，在其中“程序”文本框中输入程序名，如 FireFox，则 FireFox 就会随系统启动。

2．图形界面管理自动加载服务

选择主菜单“系统”|“系统管理”|“服务”命令，启动服务管理器，如图 4.2 所示。

该对话框列出了当前运行的服务（也就是自动运行的服务）。要取消服务，单击该服务前的对号使其为空白即可。要加载服务，使之处于选中状态即可。这个管理程序并没有提供增加

自己的服务的方法，如果要让系统加载自己的服务，需要进行额外的操作，详情将在后面介绍。



图 4.1　会话设置



图 4.2　服务设置

3．编辑脚本实现程序自动加载

前面提到了系统启动时的运行脚本，其中/etc/init.d/rc.local 脚本是系统最后一个运行的脚本。可以将运行的程序名写入该文件中，就会让程序在系统启动时自动运行。要将程序加入其中，先使用命令“sudo gedit /etc/init.d/rc.local”以 root 身份打开该脚本，在该文件最后加入一行文字，内容是“/usr/bin/gnome-terminal”。这样修改后，系统启动时就会自动运行终端程序。

4．编写自己的服务启动脚本

在编写之前，先来看一般脚本的结构（实际代码无行编号，是为了叙述方便而添加的）。该代码的好处是系统启动时运行 start 里面的代码，关机时运行 stop 里的代码，方便对该程序做额外的初始化工作。而系统关闭时也可以做些数据存储的工作，以完整关闭对于数据的读/写，避免数据的损坏或丢失。

```
1   #! /bin/sh
2   case "$1" in
3     start)
4     ·····                          #使用“start”参数执行的代码
5       ;;
6     stop)
7     ·····                          #使用“stop”参数执行的代码
8       ;;
9     restart|force-reload)
10    ·····                          #使用“restart”参数执行的代码
11      ;;
12    *)
13  exit 1
14  esac
```

这种方式对于一些数据库方面的操作尤其重要，如 MySQL 数据库。如果没有将关闭数据库的代码写入该目录，那么系统关机时关闭 MySQL 的操作就只是简单地杀掉进程。如果此时 MySQL 正在进行数据处理，就很可能会造成数据表的损坏。这种破坏，轻则造成数据库中某个表的损坏，重则数据库无法启动。如果数据库中资料比较重要，而且又没有备份或备份不够及时，就会造成极大的损失。

下面以一个简单的例子来说明如何编写自己的自启动脚本。

（1）用“sudo gedit /etc/init.d/myterminal”建立一个脚本，然后将上文示例代码输入到该脚本，其中第 4 行变为“usr/bin/gnome-terminal”，其余行内容不变。使得脚本看起来是这样：

```
2   case "$1" in
3     start)
4     /usr/bin/gnome-terminal
5       ;;
```

（2）执行“sudo chmod 755 /etc//init.d/myterminal”更改其权限为可运行。

（3）仿照系统的做法，也在 rc2.d 目录中建立脚本的符号连接，在终端中输入如下命令：

```
ln -s /etc/init.d/myterminal  /etc/rc2.d/S99myterminal
```

该文件名以 S 开头，意思是要求系统在启动时运行该脚本，后面紧跟的两位数字 99 意思是该脚本运行的优先级最低，在启动时最后运行。这些名字代表的意义参见 4.3.1 节的介绍。



系统启动时运行/etc/rcS.d 和/etc/rc2.d 目录中的脚本，将脚本放在这两个目录中任意一个，都会在系统启动时运行。不过，/etc/rcS.d 目录中的脚本一般进行的都是系统初始化的工作，如果初始化没有完成，运行用户脚本文件可能会有问题。所以，一般应该将文件放在 rc2.d 目录中。

通过如上的设置，就完成了向系统添加自运行的服务的工作。系统下次运行时会启动该脚本，然后打开一个终端。

5．使用其他工具进行管理

可用一些小工具来管理 Ubuntu 的启动选项。sysv-rc-conf 是一个不错的文字界面启动管理工具。用 sudo apt-get install sysv-rc-conf 安装这个程序，然后用 sudo sysv-rc-conf 运行，就可进行相关设置。启动后界面如图 4.3 所示。

软件第一行 service 后面的数字代表运行级别，一共有 0 到 6 再加上 s 共 8 个级别。数字下的标记代表该级别运行的程序。一般 Ubuntu 运行级别是 2，所以应着重检查数字 2 下面这一列的标记。从图 4.3 中可以看到，acpi-supps、acpid 等是自动启动的，而 alas-util 等是不自动运行的。要想改变其运行状态，按键盘的方向键，直到要更改的项，然后按空格键就可以更改其状态。注意 s 级别，这个级别是对每个级别的用户都运行的。要禁止某些程序，也可以在 s 列查看，取消相应的程序。

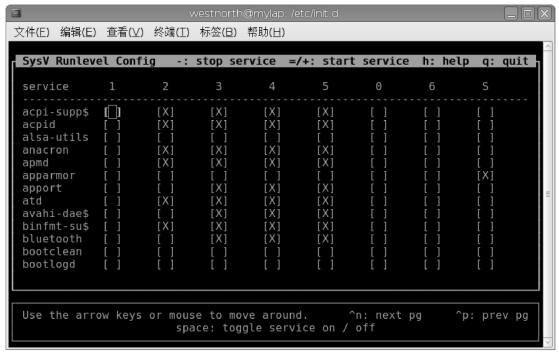


图 4.3　sysv-rc-conf 程序界面

sysv-rc-conf 还可以将脚本直接加入启动序列，例如把“/etc/init.d/myterminal”加入到系统自动启动列表中，可以使用如下命令：

```
sudo sysv-rc-conf myterminal on
```

4.3.4　常见系统服务列表

有时候，会发现系统中启动的项目很多，但却不知道这些项目都有些什么用途。在系统资源有限、想关闭某些进程时，却担心错误地关闭了系统核心进程，以致系统崩溃。为此，笔者特意整理了一个表，列出了系统中常见的进程与其功能，可供读者查阅。Linux 系统常见进程及其功能如表 4.1 所示。

表 4.1　Linux 系统常见进程及其功能

进 程 名	进 程 功 能
atd	用于进程自动调度程序 at 的守护进程
cron	用于自动调度的程序
cupsd	打印管理
dbus-daemon	消息总线系统监控进程
dbus-launch	消息总线系统执行进程
dictd	词典服务器
Fcitx	输入法
gdm	图形界面的登录管理器
getty	终端控制
gnome-brightnes	亮度控制
gnome-keyring-d	密码管理
gnome-panel	桌面系统的面板
gnome-power-man	电源管理
gnome-screensav	屏幕保护
init	系统初始进程，是所有进程的父进程
khelper	系统帮助信息进程
kjournald	日志文件系统守护进程
klogd	内核日志守护进程
kmmcd	mmc 读卡器进程
metacity	窗口管理器
nautilus	文件管理器，兼管着桌面
notification-da	桌面提示信息
pulseaudio	系统音频管理程序
scsi_eh_3	硬盘管理
seahorse-agent	PGP 密钥管理
ssh-agent	SSH 进程
syslogd	系统日志守护进程

trashapplet	回收站管理

续表

进 程 名	进 程 功 能
wicd	wicd 网络管理器进程
winbind	与 Windows 共享文件的服务 Samba 所用的一个服务
wpa_supplicant	无线网卡 WPA 认证系统
x-session-manag	X Window 进程管理
Xorg	X Window 核心程序

4.4 个性化终端

终端在 Linux 中有非常重要的作用，终端中运行的字符界面程序功能强，但占用系统的资源少，这些程序可以完成许多图形界面所无法完成的设置功能。在终端中启动图形界面程序，还可以得到图形界面的一些输出信息。如果图形界面程序出错，可以查看输出信息来确定其错误，以便解决。

终端如此重要，如果其外观或某些行为不符合用户的习惯，会给使用带来很多不便之处。幸运的是，Linux 提供了许多修改终端的界面和行为的方法，可以修改其外观、定义其行为，使之符合用户使用习惯。

4.4.1 修改终端的外观

终端的外观可以从字体、颜色配置、显示效果等方面进行配置。修改的步骤如下所示。

（1）选择“应用程序”|“附件”|“终端”命令打开终端，因为所有的这些修改项需在终端中的菜单栏进行修改。

（2）选择“编辑”|“配置文件”命令，弹出如图 4.4 所示的对话框。

（3）单击其中“编辑”按钮，弹出如图 4.5 所示的对话框，就可进行终端的相应设置了。该对话框中的各项的意义列举如下所示。



图 4.4 终端界面配置



图 4.5 编辑配置文件

- 字体设置：在“常规”选项卡中可以修改字体，一般可以选择 DejaVu Sans Mono 等宽

字体。

- 颜色设置：选择“颜色”标签，可以设置终端的显示背景和字体颜色。取消选中“使用系统主题中的颜色”复选框，在内置方案中选择“黑底绿字”选项，就可以出现像早期计算机系统中的黑底绿字效果。
- 透明背景：选择“效果”标签，在“背景”选项中，选择“透明背景”单选按钮，将滑块移动到中间位置。

（4）设置完成后，选择“关闭”按钮确认刚才的设置。

（5）关闭终端然后再启动，就可以看到黑底绿字并且带有透明效果的新终端。

4.4.2　修改终端的系统变量

有一些系统变量来制约终端的行为，这些变量可以控制终端重定向时在覆盖文件前是否提示等。可以使用 set 命令设置这些变量。set 命令有两个参数：一个是指定变量开（on）或关（off）的选项，一个是指定变量的变量名。要使某一系统变量开（有效），用-o 选项，要使其关（无效），用+o 选项。这些命令不只可以在终端中直接使用，还可将这些变量的设置写入.bashrc 文件中，以使每次启动终端时对这些变量的更改都能生效。下面是两个系统变量的意义。

- noclobber：能在重定向输出时保护已存在的文档，防止意外地覆盖一个文档。
- ignoreeof：用来禁止使用 Ctrl+D 组合键退出。

例如“set -o ignoreeof”命令的意思是关闭 ignoreeof 的变量。使用该命令后，用户只能用 Logout 或 Exit 命令退出，而不能使用 Ctrl+D 组合键来退出。

4.4.3　设置终端提示符格式

Bash 有两级用户提示符。第一级是正常情况下看到的终端中显示的内容，用户可以在其后输入命令进行相关的操作。普通用户的一级提示符是“$”，切换到 root 用户，则提示符会变为“#”。第一级提示符的系统变量名是 PS1，可以修改其来更改一级提示符的内容。二级提示符是当 Shell 期待输入更多的信息以完成命令时显示的，默认的二级提示符是“>”。二级提示符的系统变量名是 PS2。

对于 PS1 和 PS2，均可以通过一些方法来设置其显示内容。可以简单地运行命令“export PS1="I need input:"”来使一级提示符成为“I need input:”，但一般来说这样做没有什么意义，因为系统提供了许多 PS1 和 PS2 都能识别的其他有意义的特别字符，这些字符有的可以显示当前目录，有的可以显示当前时间等。

使用这些特别字符可以使得一级提示符变得更加符合自己的需求。如笔者的系统“PS1='\u@\A \w\$ '”，显示效果类似于“westnorth@16:25 ～$”。里面除了字符“@”外，其余均是特别字符。下面列出了最常用的特别字符。

- \a：ASCII 响铃字符。
- \A：显示当前时间（格式是“小时:分钟”）。
- \!：显示该命令的历史记录编号。
- \#：显示当前命令的命令编号。
- \$：显示$符作为提示符，如果是 root 用户则显示#号。
- \\：显示反斜杠。
- \d：显示当前日期。
- \h：显示主机名。

- \H：主机的全称。
- \l：当前 Shell 的终端设备名。
- \n：输出新行。
- \s：显示当前 Shell 的种类。
- \t：显示当前时间（格式是小时:分钟:秒）。
- \u：显示当前用户名。
- \W：显示当前工作目录。
- \w：显示当前工作目录的路径。

4.4.4 修改终端提示符

下面以一些实际的例子来说明如何自定义终端提示符，这些例子是直接在终端进行修改。

- 使用命令“export PS1="time is \t:" ”则终端的提示符变为“time is 09:07:48:”。export 是设置系统变量的命令，PS1 代表终端提示符，引号里的普通字符正常显示出来，而特别字符显示其特殊的内容，查看前述列表可知“\t”代表当前时间。
- 使用命令“export PS1="\u:today is \d:"”，则终端提示符变为类似“westnorth:today is Mon Oct 20:”，显示的内容是“用户名：today is 当前日期”。
- 使用命令“export PS1="\u@\A \w\$"”，则终端提示符显示“用户名@时间 当前目录”，实际终端提示符可能类似于“westnorth@09:19 /etc/esound$”。

这种方法可以比较直观、快速地设置终端提示符，但是当终端关闭后，该设置就无效了，下次需要再次设置。这种方法一般适合于单次使用的场合，如远程登录等。

4.4.5 将更改写入配置文件

要让对终端提示符的设置对于每次打开的终端都有效，可以将 export 语句加入终端配置文件“～/.bashrc”中。一般的步骤如下所示。

（1）先找出自己最理想的提示符，用 export 命令在终端中测试。

（2）使用命令“gedit ～/.bashrc”打开主目录中.bashrc 文件。

（3）将自己觉得理想的 PS1 变量设置方式，如“export PS1="\u@\A \w\$"”加入到文件的最后，保存文件。

（4）关闭终端，再打开一次就可以看到设置已经生效。

4.5 连接到互联网

现代的计算机，离开了网络就好像鱼儿离开了水，似乎毫无生气了。但 Ubuntu 一个让人不怎么满意的地方就是，看起来似乎没有一个很直观的方法可以连接到网络中。但事实并不是看起来那样的悲观，Ubuntu 提供的连网功能至少不比 Windows 更复杂。本节将分别从宽带拨号、局域网共享上网、无线上网几个方面来介绍登录网络的方法。

4.5.1 宽带拨号用户

据调查现在中国已经有 3 亿上网用户，其中大多数属于宽带用户。现在计算机已经成为许多家庭不可缺少的生活必备用品。而计算机的主要用途由前一两年前的游戏转变为上网。如果

一个操作系统不能方便地上网，无疑会让许多用户很不满意。

带着 Windows 的操作经验的用户会期待一个建立网络连接的快捷方式。从某种意义上说，在 Linux 中设置网络连接比 Windows 要容易些，而且如果建立一个可用的连接，在计算机开机后，系统就会自动连接到网络的。

1．图形界面设置宽带连接

Linux 系统提供了简便而又强大的方式用来管理网络。其中，对于初学者来说图形界面的方式是最为简单。本节介绍如何使用图形界面软件管理网络，包括宽带用户名与账号的设定等。

（1）单击桌面最上面的面板右端一个类似 的图标，选择弹出的“手动配置”命令，进入“网络设置”对话框，如图 4.6 所示。

（2）“网络设置”列表栏中有一个叫做“点对点连接”的选项，这个选项就是设置宽带连接的。单击该选项，使其选中。然后单击右侧“属性”按钮，打开“ppp0 属性”对话框，如图 4.7 所示。



对话框在打开后，可能看到里面所有的选项均是灰色、不可用的。这是因为该选项要对系统进行设置，只有 root 用户才能进行操作。只需单击“解锁”按钮，然后按提示输入用户密码，就可以进行相关设置了。

（3）单击 ppp0 属性对话框中的“启用此选项”复选框，使其选中。

（4）在“链接类型”下拉列表框中选择 PPPoE 选项，界面会变得如图 4.7 所示。

（5）在账户数据中输入用户名和密码。但是现在“确认”按钮还是灰色不可用状态。这是因为还有一个选项没有设定。

（6）选择图 4.7 所示“调制解调器”标签，进入该标签页。在“以太网界面”下拉列表框中，选择网卡，一般是 eth0，单击“确定”按钮完成设置。

（7）现在图 4.6 中“点对点连接”的状态就变为“PPPoE 以太网界面”了。这样就完成了宽带的配置，可以轻松上网了。



图 4.6　网络设置



图 4.7　宽带设置

2．命令行方式进行宽带连接

其实对于用户而言，命令行方式连接宽带其实更方便。只需要几个简单的命令就可以接入互联网，而且还可以使用 Shell 的别名功能，将这些难记的联网命令更改为简便易记的命令。

下面按顺序介绍这些拨号命令。

命令 1：pppoeconf

这个命令无论是否运行了 X Window，均可进行拨号。想象一下，如果图形界面出了问题，就可以用这个命令拨号，然后连入互联网，用 lynx、w3m 等文字模式浏览器寻找帮助了。

这个命令使用很简单，直接“sudo pppoeconf”，会出现类似于如图 4.8 所示的界面。询问找到的 eth0 网卡是否正确，选择“是”按钮。如果已经正确连接到宽带的“猫”，则会提示输入宽带的用户名和密码，按照要求输入即可。一路选择“是”按钮，完成本次拨号过程。这样就可以打开 FireFox，尽情地网上冲浪了。

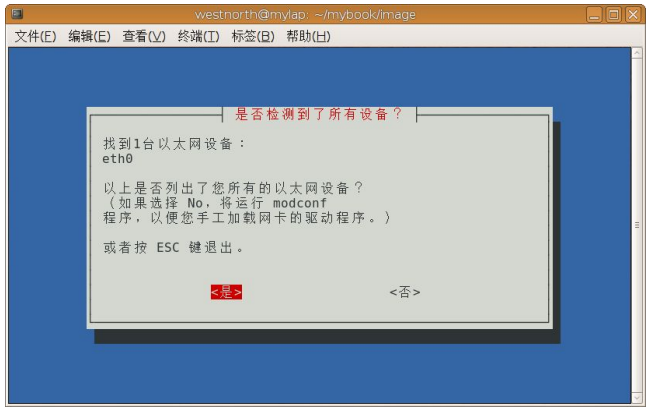


图 4.8　PPPOE 运行界面

技巧　在字符界面中可以用 Tab 键在不同的按钮间切换，如在“是”按钮与“否”按钮之间切换就可使用 Tab 键。而按下 Enter 键即可确认选择。

命令 2：pon dsl-provide

在终端中输入该命令来加载宽带连接的库文件。

注意　在运行这个命令之前，应该在图形界面或在 pppoeconf 里设置过宽带拨号的用户名和密码。运行方法是 sudo pon dsl-provider。

3．断开宽带连接

很多电信局对每个月的上网时间有限制，所以每天上网一段时间后需要断线。断线当然可以采用拔掉电话线的方式，不过更优雅的方法是使用如下命令：

```
sudo poff -a
```

poff 命令有许多的参数，如“poff -r”是重新连接，而“poff -d”则是输出宽带的调试信息。

4．获取宽带连接的信息

打 10000 号（电信服务号码）可以咨询网络连接失败的原因。在咨询时接线员可能会问：“那你的电脑获得公网的 IP 了没有？”这时候，就需要获得一些宽带联网的信息。一般来说，获取连接信息有两个命令 plog 和 ifconfig。

- plog：plog 命令显示的是宽带拨号的连接信息，如密码验证是否通过等。

- ifconfig：获取网络信息的最主要的方式。不加任何参数运行将获得本机所有网络设备的信息。也可以使用参数来限制其输出范围，如“ifconfig ppp0”查询宽带拨号的相关信息，包括本机的上网 IP（也就是 10000 接线员所说的公网 IP）、接收和发送数据的情况等相关信息。

4.5.2　局域网用户

对于使用公司的或学校网络上网的读者而言，是不存在拨号的问题的。因为网络接口上已经在源源不断地发送着海量的信息，只等将计算机连接上去，然后正确地设置就可享受互联网了。在设置之前，需要知道这几个信息，本机可用 IP 地址或可用的网段、网关地址、DNS 服务器地址等。本例假设可分配的 IP 地址是 192.168.2.22，网关是 192.168.2.2。这些信息一般可以通过咨询网管或直接查看其他机器的设定来获得。获得这些信息后，按照如下步骤进行。

（1）参照 4.5.1 节的介绍打开如图 4.6 所示对话框。本次选择“有线连接”选项，单击选中该选项。再单击“属性”按钮，打开网卡（eth0）属性对话框，如图 4.9 所示。

（2）在“配置”下拉列表框中，选择“静态 IP 地址”选项。

（3）在“IP 地址”文本框中输入一个自己在局域网中可用的地址，这一项一定不要和别人的相同。因为在一个网络中同一个 IP 地址只允许出现一次，如果设置为与其他人相同的 IP 地址，则两个人会因为 IP 地址冲突而无法上网。而且要注意的是，格式一定是“192.168.x.x”，这个地址代表局域网网址。

（4）“子网掩码”文本框的内容一般会自动生成，也就是不用手工填写。用鼠标单击该文本框，系统会自动提供一个掩码。

（5）“网关地址”文本框输入咨询或查看到的结果，一定要和同网络中其他人的一样。

设置静态 IP 后如果还是不能上网，而且在浏览器错误提示显示 DNS 服务器找不到，说明是 DNS 服务器设置错误，可参照如下步骤进行设置。

（1）打开如图 4.6 所示的对话框，单击“DNS”标签，打开 DNS 服务器设置页，如图 4.10 所示。



图 4.9　eth0 属性



图 4.10　DNS 设置

（2）单击“添加”按钮，在“DNS 服务器”中添加同一个网中其他机器设置的 DNS 服务器地址，或者可以按图 4.10 设置为 202.100.64.68。



如果要开机自动上网，“sudo gedit/etc/network/interfaces”打开该文件。检查文件内容，如果文件中没有 auto th0，在文件的最后添加该内容，则系统在启动时会自动加载该设备。

4.5.3 双机互联共享上网

在实际应用中，有这样的情形，家庭里有两台计算机。一般情况下，一台是台式机，另一台是笔记本，然后想要两台计算机共享同一个宽带。其实这种方式的上网也很容易实现，照着下面的介绍来做就没问题了。

1．需要的硬件

两台计算机，一台要有网卡（从机），另外一台要有两个网卡（主机）、一条宽带接入线、一条用于双机互联的网线（注意两条网线接口是不同的，连接 ADSL 是正接的，双机互联是交叉的，在电脑店里接网线时告诉老板是双机互联，老板就会选择正确的接线方式）。

2．主机的设置

先介绍主机的设置。主机有两个网卡，设其为 eth0 与 eth1，而 eth0 接到了宽带线上，eth1 接在从机上。再次打开如图 4.6 所示的对话框，设置各个网卡的 IP 地址。

- eth0（连接到宽带“猫”）的网卡 IP 设置为 192.168.1.2，掩码为 255.255.255.0，网关为 92.168.1.1。
- eth1（连接到从机）的网卡 IP 设置为 192.168.2.1，掩码为 55.255.255.0，网关为空。

3．从机的设置

继续在如图 4.6 所示的对话框中进行设置。从机的网卡 IP 设置为 192.168.2.2，掩码为 55.255.255.0，网关为 192.168.2.1（主机连接到从机的网卡的 IP）。



主机中 eth0、eth1 的 IP 地址最好不要在同一个网段上。这个网段，是指 IP 地址中第 3 组数字。如可将第 3 组数字设置为一个为 1，一个为 2。

4．检查网络连接情况

接下来就是插上网线，检查网络连接了。先在主机的终端运行“ping 192.168.2.2”，检查是否和从机连通。如果得到的输出有 0.128ms 之类的信息，说明双机连通了。如果输出是“unknown host address”之类的信息，说明网络不通。排除其他原因（如网线质量问题等）后仍然不通。则一个可能的原因是 Windows 会在启动时将网卡的属性修改，就需要到 Windows 中做相关设置。具体步骤如下。

（1）进入 Windows 操作系统，如 Windows XP。

（2）在桌面“我的电脑”图标上右击鼠标，在弹出的菜单中选择“属性”命令。

（3）在弹出的对话框中选择“硬件”标签页。在标签页中单击“设备管理器”按钮，出现硬件列表，在网络设备中找到网卡，双击打开其属性。

（4）在出现的对话框中单击“高级”标签，打开高级设置标签页，如图 4.11 所示。

（5）单击属性列表框的“Wakeup on Link Change”项，然后在“值”下拉框中选择“Enable”项。

现在重新启动 Linux 操作系统，网卡应该可以正常使用了。



图 4.11　网卡高级选项

5．设置

最后在主机中再进行一些最重要的设置，完成这些设置后从机应可以通过主机的连接正常上网了，这些设置一共有两步。

（1）运行“sudo gedit /etc/sysctl.conf”，打开系统设置文件，在文件后添加一行“net/ipv4/ip_forward=1”。

（2）运行“sudo gedit /etc/initd/rc.local”，打开启动脚本，在其中添加一行“sudo iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o ppp+ -j MASQUERADE”。

上述都做好后，从机应能上网了。如果不行，可能还需要对 iptables 做一些操作，才可解决部分从机不能上网的问题。这些操作的具体步骤如下所示（在终端中依次运行如下命令）：

```
$sudo iptables -F
$sudo iptables -P INPUT ACCEPT
$sudo iptables -P FORWARD ACCEPT
$sudo iptables -t nat -A POSTROUTING -o etho -j MASQUERADE
```

4.5.4　无线上网

在小型应用环境下，无线 AP 配置无线网卡共享上网的方式比较普遍。这是由于无线网络具有不用布线、移动灵活的特点。但 Ubuntu 在使用无线网卡上网时遇到的困难也比较多，如驱动的问题、WEP 加密的问题等。这里简单地介绍如何解决无线网卡驱动的方法。

Linux 中无线网上驱动可以选择 Madwifi 或 Ndiswrapper。这些程序的原理是提供一个模拟层，用来将 Windows 中的驱动程序转换为 Ubuntu 可识别的格式。但 Madwifi 可能对于某些网卡的支持度并不是很理想，就可以选择 Ndiswrapper。Ndiswrapper 安装网卡驱动的过程如下所示。

（1）运行“sudo apt-get install ndiswrapper-* ”，或者在新立得里搜索 ndiswrapper，安装搜索出的结果。安装成功后，就可以用其安装网卡驱动了。

（2）将 Windows 版的无线网卡驱动复制到一个目录，如～/driver/wireless.inf。

（3）运行“sudo ndiswrapper -i wire.inf”安装驱动。

（4）使用这个命令来显示当前的驱动“ndiswrapper -l”，可以看到驱动已经出现在列表中。

（5）使用命令“sudo gedit /etc/modules”，在文件末尾加入 ndiswraper。意思是系统启动时加载 ndiswrappe 模块。

（6）执行“sudo depmod -a”加载 ndiswrapper 模块。

（7）执行“sudo ndiswrapper -m”将信息写入/etc/modprobe.d/ndiswrapper。

（8）使用命令“sudo gedit /etc/modprobe.d/blacklist”，在文件末尾加入“blacklist zd1211rw”，zd1211rw 是网卡芯片，将该行加入黑名单，意思是不让系统自动加载该驱动，而是使用 ndiswrapper 的驱动。

经过如上设置，Linux 会在启动时自动加载相应的驱动程序。在插入无线网卡后，系统会通过 ndiswrapper 加载 Windows 的驱动，然后就可正常使用网络了。

4.6 中文化

对于中国使用者来说，Linux 的英文界面还是让许多人望而却步。虽然现在各发行版的界面基本都可以中文化，但实际使用中重要的 man 手册，仍然是纯英文的，这不能不让人感到遗憾。本节并不是介绍如何中文 man 文档，而是介绍了一些中文化的概念，以及安装中文支持的方法。

4.6.1 中文化概述

很多时候，都可以听到一些中文化的传闻，比如“第一款中文内核的操作系统”之类。但有时运行才发现，这些号称中文内核的操作系统的帮助信息都是英文信息，就让人有点啼笑皆非了。每种语言其实均有其相应的国际标准，如果符合标准，就会给对外交流带来很大的方便。如若不然，会给自己的语言国际化带来困难，也阻碍了与外界的交流。一般来说，中文的 Linux 要包括如下方面。

1. 中文 Locale 支持

如果查看/usr/share/xsession/gnome.desktop 文件，会发现里面有类似这样的信息。

```
Comment[tr]=Bu oturum ile GNOME'a giri yaparsnz
Comment[uk]=С е а н с  р о б о т и  в  с е р е д о в и щ GNOME
Comment[vi]=Session này cho bn ng nhp vào GNOME
Comment[zh_CN]=此会话使您登录到 GNOME
Comment[zh_TW]=选取这个作业阶段后会进入 GNOME 环境
```

也就是说，一些国际化程度较高的程序里，会自动判断相应的 Locale，然后输出相对应的信息。对于中文而言，如果当前 Locale 设置为 zh_CN，要能正确显示“此会话使您登录到 GNOME”才能称得上是一个中文化的系统。

2. 多内码支持

由于中国内部就有 GB 和 Big5 两种编码，中文支持就要能正确支持多内码。

3. 输入支持

比如目前使用极广泛的 Scim 输入法就是一款支持 GB 和 Big5 的输入法。

4．中文打印支持

现在一般的软件均可实现，但还有个别软件对打印支持不是很好，如 vi 系列（vi、vim、gvim）。

5．man 手册等文档的中文化

这是当前 Linux 发展中最为薄弱的一个环节，到目前为止，仍然没有一个比较完整的 man 手册。

对于 Ubuntu 而言，除了第 5 条，其余均可以做到完美的支持。

4.6.2　安装中文支持

对于中国使用者而言，不管是什么操作系统，首先看重的是界面的中文化。如果 Ubuntu 安装时选择了中文界面，安装后应该界面有部分中文界面（使用 CD 版安装）或全部是中文界面（使用 DVD 版安装）。对于 CD 安装的用户，要实现完整的中文界面，也很容易。首先要保证已经联上网，因为语言支持需要到互联网上下载许多内容。然后按如下步骤进行即可：

（1）选择主菜单“System”（系统）|“Administration”（系统管理）|“Language Support”（语言支持），出现如图 4.1 所示对话框。

（2）在“Supported Language”（支持的语言）列表框里选中“Chinese”选项（也就是选择其后的“Support”复选框）。

（3）在“Default Language”（默认语言）下拉框里，选择“Chinese（China）”选项。

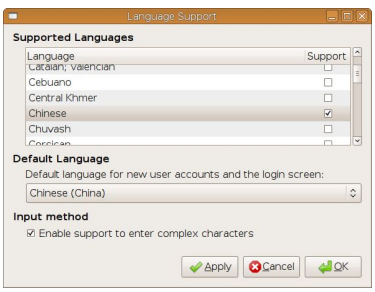


图 4.12　“语言支持对话框”

（4）选中“Enable support to enter complex characters”（支持复杂语言输入法）复选框。

（5）单击“Apply”（应用）按钮应用所选项目。这时会在网络中下载所需内容。由于中文支持包比较大，下载可能需要一段时间。耐心等待一会，下载完成后，重新启动计算机，就可以看到所有界面均变更为中文。

4.7　安装中文输入法

Linux 中常用的中文输入法有 Scim 和 Fcitx 两种。其实这两种确切地说应该称为输入平台。因为每个平台均含有几种常见的输入法，如拼音、五笔等。相对来说，Scim 更为国际化一些，支持中、日、韩等文字的输入，而 Fcitx 则比较小巧精致些。下面分别介绍其安装与使用技巧。

4.7.1　Fcitx 输入法

Fcitx 是一款优秀而又小巧的中文输入法。在很长一段时间里，Fcitx 由于种种原因停止了更新。但从 2008 年 10 月开始，该软件又重新开始了更新，而且新增加了一款“五笔拼音输入法”，比较好用。

1．安装 Fcitx

安装 Fcitx 是很简单的。因为软件库中有该输入法，只要使用命令“sudo apt-get install fcitx”就会自动完成安装。

2．设置

安装后，还需进行相关的设置，才能正常使用。设置的步骤如下。

（1）使用命令“sudo /etc/X11/xinit/xinput.d/fcitx”建立一个文件，输入如下内容：

```
XIM=fcitx
XIM_PROGRAM=/usr/bin/fcitx
XIM_ARGS=""
GTK_IM_MODULE=XIM
QT_IM_MODULE=XIM
DEPENDS="fcitx"
```

（2）将 root 用户的输入法改为 Fcitx。

```
sudo im-switch -a fcitx
```

（3）将普通用户的输入法改为 Fcitx。

```
im-switch -a fcitx
```

3．精简 Fcitx

Fcitx 提供了许多输入法，有许多是普通用户不可能去使用的。如果觉得这些输入法太多影响使用的方便程度，就可以去除一些不用的输入法。方法很简单，Fcitx 的设置文件均是中文，比较易懂。只需打开下列两个文件，然后进行相应的修改即可。

- 打开“～/.fcitx/config”文件，删除不需要的码表。
- “sudo gedit /usr/share/fcitx/data/tables.conf”，以 root 用户打开输入法码表文件，删除里面不需要的码表。

4．修改 Fcitx 的字体

运行 gedit ～/.fcitx/config，将字体大小的数字调整到自己满意为止。

5．中文显示的问题

有时，安装了 Fcitx 后，状态栏显示的不是汉字，而是一个个方框。这种情况可能是由于没有中文支持环境，其解决办法如下。

（1）使用命令“sudo gvim /var/lib/Locales/supported.d/zh”新建一个文件。

（2）在该文件中添加如下内容：

```
zh_SG.UTF-8 UTF-8
zh_HK.UTF-8 UTF-8
zh_TW.UTF-8 UTF-8
zh_CN.UTF-8 UTF-8
zh_CN.GB2312 GB2312
zh_CN.GBK GBK
zh_CN.GB18030 GB18030
zh_HK.Big5 Big5
zh_TW.Big5 Big5
```

（3）使用 sudo Locale-gen 命令创建中文环境。

经过上面 3 步，就建立了中文环境。一般应该解决问题。



如果建立了中文环境，Fcitx 显示还是一个个方块，那就应该是中文字体的设置问题了。打开“~/.fcitx/config”，设置如下内容“显示字体（中）=wenquanyi zenhei”一般可以解决问题。

6．应用配置文件

Fcitx 修改配置后按 Ctrl+5 组合键即可重新加载新的配置文件。

4.7.2　Scim 输入法

Scim 是 Ubuntu 系统默认的中文输入平台，当然作为 Ubuntu 的默认中文输入平台，其是有许多优点的。

- 稳定：可以在更多的程序中正确输入。
- 国际化：支持 Big5、日、韩等国输入。
- 设置方便：直接在界面上单击右键就可设置，而不像 Fcitx 要更改配置文件。
- 使用方便：很重要的一点是 Ubuntu 发行时，将其作为默认的输入法，只需安装自己常用输入法的码表即可。

安装 Scim 也比较简单，只需运行“sudo apt-get install scim scim-pinyin scim-tables-zh im-switch scim-qtimm scim-bridge scim-bridge-client-gtk scim-bridge-client-qt scim-bridge- agent”，就将 Scim 安装在系统中了。而且如果安装 DVD 版的 Ubuntu，则系统会自动安装好 Scim 输入法，直接就可使用。

前面提到过，Scim 的设置比较简单，在桌面的右上角右击代表 Scim 输入法的键盘图标，在弹出的快捷菜单中选择“Scim 设置”命令，出现的对话框如图 4.13 所示。在左侧的列表框中选择“全局设置”选项，取消右面如“繁体中文”和“其他语言”的选项，只保留常用的输入法，如“智能拼音”、“Wubi”（五笔）等。单击“确定”按钮，保存设置并退出。现在就可以通过按“Ctrl+空格”组合键来切换到 Scim 输入法并开始中文输入。

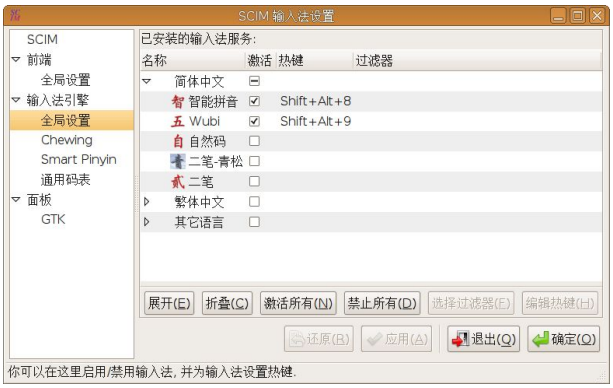


图 4.13　Scim 设置对话框

1．添加用户词组

用户在输入的时候，可能会频繁输入一些特定的词，如公司名称、地名等，但很可能这些

词组并没有收录到 Scim 中。这时，就可以使用自定义词组功能来提高输入效率。Scim 定义词组的方法比较特别，具体步骤如下。

（1）按下 Esc 键，清空 Scim 输入缓冲区中文字内容。

（2）在任意文本编辑器中输入要加入 Scim 词库中插件词组，如插件。

（3）按下 Ctrl+A 组合键指示 Scim 添加刚才输入插件词组，Scim 会提示输入该词组插件编码，可以自由输入，不一定要符合五笔编码规范。

（4）按下 Enter 键，Scim 应该会提示 sucess，意思是加入词组成功。

2．添加多个用户词组

如果觉得以上添加词组的方法效率比较低，也可以采用编写配置文件的方法来添加词组。添加步骤的具体步骤如下。

（1）打开文件“～/.scim/sys-tablels/Wubi.bin.user”。

（2）在 BEGIN_TABLE 后添加要自定义的词，如下所示：

```
BEGIN_TABLE
rtwr    插件    0
rtwa    其他插件  0
END_TABLE
```

4.8 Locale 简介

Locale 的英文意思是地区或者地域。但在 Linux 环境里，这个单词所代表的意义远不至此。Locale 定义了软件界面的语言，货币样式、日期格式等信息。其是 Linux 支持多语种的重要基础，许多软件会根据 Locale 的设置来显示相关的语言界面。

4.8.1 Locale 内容详解

使用命令 Locale 查看本机的设置，输出内容应该类似于：

```
LANG=zh_CN.UTF-8
LC_CTYPE="zh_CN.UTF-8"
LC_NUMERIC="zh_CN.UTF-8"
LC_TIME="zh_CN.UTF-8"
LC_COLLATE="zh_CN.UTF-8"
LC_MONETARY="zh_CN.UTF-8"
LC_MESSAGES="zh_CN.UTF-8"
LC_PAPER="zh_CN.UTF-8"
LC_NAME="zh_CN.UTF-8"
LC_ADDRESS="zh_CN.UTF-8"
LC_TELEPHONE="zh_CN.UTF-8"
LC_MEASUREMENT="zh_CN.UTF-8"
LC_IDENTIFICATION="zh_CN.UTF-8"
LC_ALL=
```

这个内容可能会让绝大多数读者一头雾水，其实里面的内容也不难。分析一下，会发现其

主要分为两部分。

1．Locale 第一部分

Locale 的第一部分即等号左边的内容。这部分内容是 Locale 所设置的项目内容，如数字日期显示格式、电话显示格式、度量单位显示格式等。表 4.2 说明了其意义。

表 4.2　Locale 第一部分所代表的意义

Locale 第一部分的项目	该项目所代表的意义
LC_CTYPE	用户所使用的语言符号及其分类
LC_NUMERIC	数字显示格式
LC_COLLATE	比较和排序习惯
LC_TIME	时间显示格式
LC_MONETARY	货币显示格式
LC_MESSAGES	提示信息格式
LC_NAME	姓名显示格式
LC_ADDRESS	地址书写方式
LC_MEASUREMENT	度量单位（如长度、温度等）显示格式
LC_PAPER	纸张大小
LC_IDENTIFICATION	Locale 对自身包含信息的概述

要注意的是这些项目是彼此独立的，可以按照不同的需要，在同一个系统内设置其为不同的值。也就是说可以同时使用美国的日期格式，欧洲的数字格式、中国的度量模式等。例如可以把 LC_CTYPE 设定成 zh_CN.GB2312，而其他所有的项都是 en_US.UTF-8。

上面对 Locale 各项内容解释中，并没有提及 LANG 和 LC_ALL。这并不是说其不重要，而是这两个选项有其特殊的意义，所以要单独介绍。

- LC_ALL 发布号令“其他的 Locale 都设置为我的样式”，然后所有的项均会设为其样式。如 LC_ALL=zh_CN.UTF-8，则所有的项不管原来的内容是什么，均强制性更改为 zh_CN.UTF-8。
- 而 LANG 发布公告“建议大家设置为我的样式”。如 LANG=en_US.UTF8，则那些没有显式设置内容的 Locale 设置为 en_US.UTF-8，而原来已经设置了内容的 Locale 其内容不变。

2．Locale 第二部分

再来看看等号后面的内容 zh_CN.UTF-8。zh_CN 意思是国别。如 zh 代表中国，而 CN 代表中国大陆。而如果是 zh_HK，就代表中国香港。UTF-8 代表编码设置，每一个国家、地区均可设置不同的编码。如可设置为 zh_CN.GB2312 代表中国大陆而编码设置为 GB2312。

4.8.2　Locale 的作用及定义文件

知道 Locale 各项代表的意义，就明白了 Locale 的作用。如设置 LC_MONETARY=zh_CN.UTF-8，则系统在显示货币格式时就会显示 zh_CN 定义的格式，可能是￥12.35。而如果使用 LC_MONETARY=en_US.UTF-8 设置，则可能显示$12.35。可以说，Locale 就是某一个地域内人们的语言习惯和文化传统，以及生活习惯。

Locale 定义文件在/usr/share/i18n/locale 目录中。每一个 Locale 都是一个文件，如 zh_CN、

en_US 等。在这些文件中定义了具体的实现方式。如果对其内容比较精通，还可以自行修改以符合自己的要求。

用 ls 查看/usr/share/i18n/locale 里的文件，会发现里面有些文件名称类似于 it_IT@euro。这个@代表修正项。如果两个 Locale 中大多数内容一致，只有个别不同，就可以不用写一个新的 Locale 文件，直接修正原来的就可以了。

打开 it_IT@euro，可以看到很多类似这样的结构。

```
LC_MEASUREMENT
copy "it_IT"
END LC_MEASUREMENT
```

这说明 Locale 中大部分内容直接来源于 it_IT。文件中也实现了自己的内容，如 LC_IDENTIFICATION，查看上表可知新的意大利 Locale 和原来的只有 Locale 对自身的描述不同。

4.8.3 设置 Locale

通过上面的介绍，现在应该对 Locale 有些初步了解了。在实际使用中，设置正确的 Locale 会为使用带来很大的便利，下面简单介绍一些设置 Locale 的技巧。

- 对于大多数用户而言，可以设置 LC_ALL=zh_CN.UTF-8，强制设置所有的项为 zh_CN.UTF-8。这就是一个纯中文的系统（包括数字格式、日期格式、货币格式等均符合中国的习惯）。
- 有的用户也许只想在系统中输入中文，而对于界面是否为中文并不在意。因为设置为英文可能会获得更好的性能。经过实际的测试，设置为中文的 Locale 会比英文的 Locale 有一定的性能损失。这种情况就可以只设置 LC_CTYPE = zh_CN.UTF-8，LANG=en_US.UTF-8 就可以了。这种设置方式，只要安装了前面所介绍的中文支持，则无论 Locale 怎样设置均可正确输入中文。
- 如果仅仅需要在某个程序里更改 Locale 设定，可用一个命令暂时修改 Locale 设定。如类似这样的命令“env LANG=zh_CN.UTF-8 gedit”，会以中文的界面打开文本编辑器 gedit，而其他的程序界面仍显示为原来的 Locale 设置。

4.9 小结

本章的内容在本书中也是极为重要的一章，这是由于 Linux 的发行版还远未达到完美的水平，有许多用户常用的功能还未能满足，这就需要进行系统层级的设置，以使之基本达到可用的目的。而本章介绍的就是这些相关的设置。包括如何连接到网络、如何安装输入法、如何管理系统启动项目与服务等实用功能。本章的难点内容是理解 Linux 的启动过程。要理解启动过程，就应该参照书中的提示，依次查看、分析相关的脚本文件与配置文件。而理解了启动过程，会为管理启动项目与进行系统设置带来很大的方便。

第 5 章　文本编辑器与正则表达式

文本编辑器的功能是编辑纯文本格式文件。其功能重心并不是排出漂亮的版面，而且不能在其中绘制表格，当然也不能在其中插入图片等。但这并不意味着文本编辑器没有用武之地，因为其对文本内容的处理效率要远高于那些办公套件中所谓的文字处理软件。所以文本编辑器在 Linux 的系统管理和对于特定格式文件的处理中发挥了极其重要的作用。而为了对文本文件进行高效地操作，正则表达式的知识又是不可或缺的，所以本章也将介绍些简单的正则表达式的知识。

5.1　Ubuntu 默认的文本编辑器 gedit

Ubuntu 系统中默认安装了一款看似普通的文本编辑器，其名字是 gedit。这个编辑器被许多用户使用过。对于大多数使用者来说，这只是一款很普通的甚至有些简陋的文字编辑软件，有点像 Windows 环境中的记事本。但其实经过一些简单的设置，其也可成为功能强大的编辑器甚至是编程的高级 IDE。通过本节的介绍，读者可重新认识到其功能的强大。

5.1.1　程序界面

程序运行的方法很简单，依次选择“应用程序”|“附件”|“文本编辑器”命令或简单地在终端中输入 gedit 就可将其打开。默认情况下，其界面极其普通。但实际上初始情况下的 gedit 仅仅发挥了其强大功能的极小部分。像大多数开源软件一样，其开放了插件接口，通过插件扩充功能后，其不亚于任何一款文本编辑器。

对于 Windows 用户来说，其还有一些很重要的优势：图形界面简洁直观、上手快、不用记忆命令就可方便地进行操作，同时文本编辑区域支持鼠标，可以对文本进行选择、拖动等操作，符合其他软件中养成的操作习惯。

先来看一张安装了 LaTeX 插件的 gedit 界面，如图 5.1 所示。这张图应该可以改变许多用户认为其功能简单的想法了。从图 5.1 中可以看到 gedit 的部分功能。

- 语法着色：就是可以根据不同的代码，将不同的关键字用不同的颜色显示出来。这样，编辑时可以直观地反映出文件的某些结构，方便编辑。
- 文本行编号：给文本编号，似乎没什么意义，但实际上在代码编写中，显示代码的行号对于程序编写是很有用处的，如“将第 11 行到 20 行移到 32 行之后”等功能就依赖于行号。
- 代码自动补全：输入代码时，只需要输入函数或命令的前几个字符，系统列出以这几个字符开始的所有函数列表，以供用户使用。这就是代码的自动补全或自动提示功能。有

了该功能，编写代码时可大幅度提高工作效率。

- 内嵌终端：gedit 可以在界面下方显示一个终端窗口，这样在需要在终端进行某些工作时，就可以不离开编辑器，直接在该窗口中操作。该终端功能与从主菜单打开的终端窗口功能完全一致，可以完成任何在系统终端中能够完成的操作。
- 特殊功能工具栏：在打开特定的文件时，可以显示特定的工具栏，完成特定的功能。如安装了 LaTeX 插件后，打开 TeX 文件，就会自动显示 LaTeX 处理的相关工具栏，可以完成 TeX 文件的编译，插入 TeX 的特定结构等。

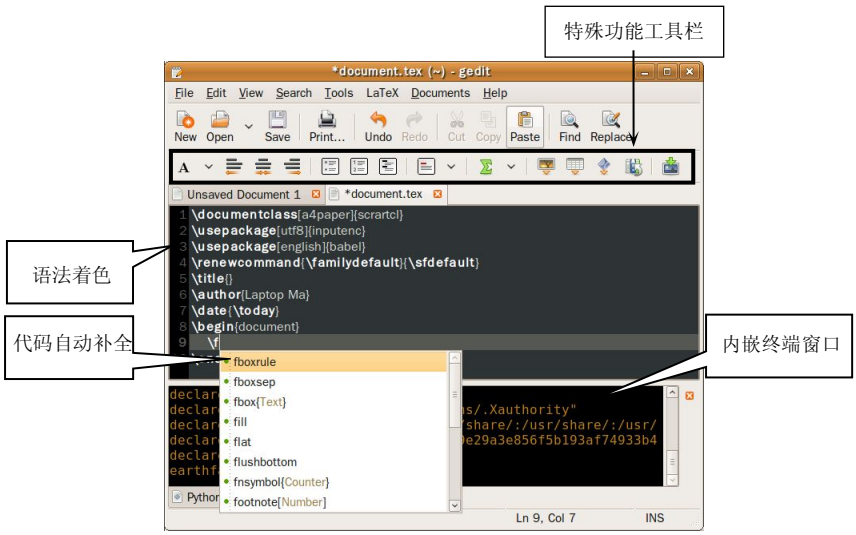


图 5.1　gedit 界面

上面的一些特点可以很直观地从图 5.1 中看出来。但是 gedit 中还有一些特色功能是隐藏在菜单中的，需要打开菜单来发现。其中菜单 Edit 里有以下几项特色功能。

- Join Lines：连接行，意思将光标所在行和下一行合并为一行。由于 Linux 系统的许多文本编辑器会将文本按 80 个字符切断，造成使用上的不方便。使用该功能，可以简单地将这些被分割的行连接起来。
- Split Lines：分割行，和上面的一个命令刚好相反。
- Indent：缩进，将一行的文本向后缩进一些宽度。
- Unindent：反缩进，将一行的文本向前伸出一些宽度。
- Tool 菜单里有一个统计字数的功能，即 Document Staticstics 菜单项。该菜单项可以统计文本中的行数、字数等信息。



这些菜单的功能远不是 gedit 功能的全部，从 gedit 的官方网站可以下载 gedit 更详细的功能与特色列表。其官方网址是 http://www.gnome.org/projects/gedit/。

5.1.2　基本设置

显然，初始情况下的 gedit 并没有如图 5.1 所示的强大。但是，千里之行，始于足下，gedit 提供了良好的基础，可以从该基础上建设一个强大的编辑环境。gedit 的基本设置均可以通过依次选择 Edit | Preferences 命令来实现。选择 Edit | Preferences 命令，打开如图 5.2 所示的对话框。

该对话框用来设置的主界面，下面列出了各项的意义。

- Text Wrapping 选项：用于设置字绕回的方式。所谓字绕回是如果文本的长度大于程序显示的宽度，会自动将文本分行显示。如果取消选中 Enabl text wrapping 复选框，也就是取消了字绕回功能。这样，无论文本行有多长，均只显示在程序的一行中，如果要查看文本行的其他内容，用户必须拖动窗口底部的滑块。
- Line Numbers 选项：设置是否显示文本行的编号。当选择 Display line number 复选框后，会在窗口左侧显示文本行的编号。但这些编号并不实际存入文件内容中。
- Current Line 选项：当前行，如果选择 Highlight current line 复选框，就将鼠标所在的行高亮显示。这样方便定位鼠标的位置。
- Right margin 选项：右边界，设置是否显示右边界。如果选择 Diplay right margin 复选框，则在页面的右侧显示一条标志页面宽度的垂直线，线的位置由 Right margin at column 微调按钮指定。普通用户一般不用设置该选项。
- Bracket Matching 选项：匹配括号。这也是一个很有用的选项。如果选择 Highlight matching bracket 复选框，在文本中将鼠标放置在一个括号前时，就会自动高亮显示与该括号匹配的另一半括号。这项功能可以用来检查代码中的括号是否匹配。

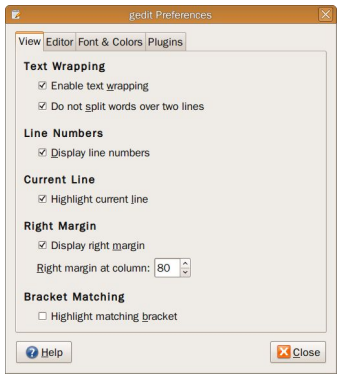


图 5.2　gedit 基本设置

选择 Editor 标签，就可以在 Editor 选项卡中设置 Tab 键的宽度、是否自动缩进、自动保存等选项。而选择 Font & Color 标签，就可在其中设置软件使用的字体，还可以设置软件显示文本时背景的颜色、文本的颜色等。

5.1.3　安装插件

插件对于 gedit 的重要程度，就像武器对于军人的重要程度一样。gedit 的强大，很大程度上在于其拥有许多强大的插件。软件在安装时，也附带了许多插件，但这些插件还是远远不够的，许多用户均需要安装自己的插件扩充 gedit 的功能。本部分内容主要是设置 gedit 自带的插件和安装新插件。

1．设置自带的插件

Ubuntu 的软件源文件中有几个 gedit 基本功能的插件，在安装其他插件前应该先安装这些基本插件。安装可以使用如下命令。

```
sudo apt-get install gedit-plugins
```

安装成功后，选择“Edit”|“Preferences”命令，弹出如图 5.3 所示对话框。选择 Plugins 标签，切换到插件标签页，可以看到其中多了许多插件，界面如图 5.3 所示。

选择插件前面的复选框，即可启用该插件。一般来说，应该启用的插件有如下几项。

- Tag list：标记列表，启用这个插件可以开启代码自动补

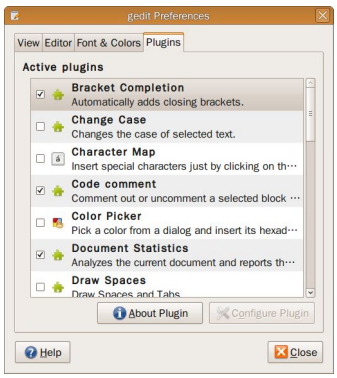


图 5.3　插件

齐功能。

- Embedded Termined：内嵌终端。启用这个插件时可以在 gedit 编辑窗口下侧显示一个终端窗口，其实就是内嵌了一个 Bash。
- Snippets：代码片断。启用该插件可以在输入程序语句时自动补全该命令。如在 C 语言源代码中输入 while，按下 Ctrl＋空格键自动补全，就会自动补上 while 结构的完整语句，如下所示：

```
while (condition)
{
}
```

- File Browser Pane：在侧栏开启文件目录浏览功能，方便访问系统中目录与文件。
- Bracket Completion：自动关闭括号，如果在文档中输入一个左括号，系统自动加上右括号。
- Code comment：代码注释功能。根据文档类型自动将选中的文本变为注释语句。
- Document Statictics：文档统计功能，可以统计出文档的字数、行数等信息。

2．安装其他插件

很多时候，gedit 自带的这些插件还不能满足使用时的选定要求，就可以安装特定的插件。其实安装的方法也很简单。本例以 LaTeX 编辑插件来说明一般插件的安装过程。

（1）LaTeX 插件使用了 rubber，如果要 LaTeX 能正常使用，需要先安装 rubber。方法很简单，在终端运行“sudo apt-get install rubber”即可。这一步对于大多数插件是不需要进行的。

（2）LaTeX 的 gedit 插件位于 http://live.gnome.org/Gedit/LaTeXPlugin。下载后会得到一个压缩文件，将文件解压，会得到一个名为 LaTeXPlugin-0.1.3.2 的目录。

（3）打开该目录，将 plugins 目录中的 LaTeXPlugin 目录及其中的文件复制到～/.gnome2/gedit/plugins 目录。如果目标目录不存在，就建立该目录。

（4）将 LaTeXPlugin.gedit-plugin 文件也复制到～/.gnome2/gedit/plugins 目录。

一般来说，要安装插件，只需将插件文件复制到～/.gnome2/gedit/plugins 目录即可。本次安装，该目录中的文件结构如图 5.4 所示。



图 5.4　插件目录中的文件

（5）插件安装后，还需要选择 Edit|Preferences 命令，然后选择 Plugins 标签，切换到插件标签页，在其中启用 LaTeX 插件，就可以在 gedit 中编辑 LaTeX 文件了。

5.2　瑞士军刀 vim

vi 是 UNIX 及类 UNIX 世界中最为优秀的文本编辑器之一，另一款则是 Emacs。虽然 vi 上手有点困难，但熟练使用 vi 的用户均对其优秀的性能赞不绝口，而且 vi 是几乎所有类 UNIX

系统中默认安装的文本编辑器。如果工作在类 UNIX 环境，不用担心学习 vi 的经验被浪费。从个人电脑到大型机均以同样的方式使用 vi。在很多情况下，vi 还是唯一的选择。如进行远程登录维护时，可用的文本编辑器就只有 vi 了。vim 是 vi 的加强版，拥有 vi 的全部功能，本节介绍的内容就以 vim 为基础。

5.2.1　初识 vim

美国加州大学伯克利分校大学生 Bill Joy 在 1976 年开发了 vi。所以 vi 距今已经有 30 多年历史，比许多使用者的年龄还要大。vi 在开发初期是一款文本行编辑器的可视化接口（vimsual interface），并因此而得名。

vi 有个特点，在文本中移动光标位置不推荐用键盘上的方向键，而推荐使用“j、k、h、l”等字母键。这种情况的出现有两个原因，一是与其历史有关，其早期是在打字机终端上用的，而打字机上是没有方向键的；二是这种方式有着操作的快捷，使用字母键移动光标手不用从打字区离开，效率会高很多。

随着 vi 的使用者越来越多，出现了许多对于 vi 功能进行扩展的软件。现在最为流行的也最为强大的非 vim 莫属了。vim 是由 Bram Moolenaar 开发的。vim 的含义是 Vi Improved（vi 的增强版）。为了适应初学者的需求，又开发了 vim 的图形界面，就是 vim-gnome，一般简称为 gvim。相对来说，gvim 算是用户友好的，其拥有了菜单、工具栏，不需要记住所有的命令即可进行相关的操作。而且 gvim 还保持了 vi 的强大性能，所以也得到了较为广泛的应用。

纯粹的 vim 必须在终端中输入 vim 来进入，因为其是屏幕编辑软件，只拥有字符界面。其界面的主要特点是没有工具栏、没有菜单、不能使用鼠标，一切要依靠用户学习得来的键盘命令来实现对 vim 的操作。启动后的 vim 典型界面如图 5.5 所示。



图 5.5　vim 的界面

面对这样一个窗口，初学者无疑会一头雾水。看起来这似乎是从原始的计算机世界里重新发掘出来的古董，无法进行文件操作，如文件的打开、新建等，而且最离奇的是，这个号称是文本编辑器的程序无法输入一个文字。

这就要介绍一下 vim 的模式了。vim 启动后默认是命令模式，也就是说会将用户所有的键盘输入当做命令来执行。在命令模式中输入字母 i（或 o、a）就可进入编辑模式，进行正常的文本输入和编辑了。vim 和其他文本编辑器不同的是，它是一款有模式状态的编辑器，其一共拥有四种模式。

- 命令模式：该模式中所有输入均被当做命令执行，而不是作为文本输入。可以按下 i、o、a 键进入编辑模式。

- 编辑模式：像其他文本编辑器一样的模式，在该模式中可以正常进行文本的输入、编辑。按下 Esc 键退出编辑模式，进入命令模式。因此也可以说其他文本编辑器（如 gedit）只有一种模式。对于编辑模式中的细节，本书的读者应该具备这方面的基本知识，本节就不予介绍了。
- 可视模式：当某些文本被选中后进入该模式，可以进行文本块的复制、粘贴等操作。
- 命令行模式：在命令模式时，按下：（就是同时按下 Shift＋；键）便会在窗口底端出现一个用于输入命令的行，一般称这种状态为命令行模式。



vim 的命令行模式很强大，可以发出类似这样“将第 3 行到第 28 行中包含 test 的行全部删除”、将“第 38 到 49 行内容移动到第 108 行之后”等之类的命令，还可以调用系统命令完成诸如“将文件的第 2 行到第 1000 行按第四列排序”的功能。

先来看看如何在 vim 中如何进行文件操作，这对于大多数用户来说是首先应该知道的知识。

- 打开文件：按下 Esc 键确保处于命令模式，然后按下：键，进入到命令行模式。打开文件的命令是“:edit”（edit 可以理解为在 vim 中编辑一个文件，也可以使用其简写形式“:w”）。如要打开主目录中的 test 文件，就可用“:e ～/test”。打开文件后，按下 i 键进入编辑模式，就可进行一些文字编辑。
- 保存文件：文件修改后，应该记得先将其保存。保存的命令是“:write”（write 的意思是将对文件的修改写入磁盘）。按下 Esc 键进入命令模式后，按下：键进入命令行模式，然后输入 write 或者简单地输入 w 即可完成文件的保存。
- 退出：同样进入命令行模式，然后输入 quit 或简单地输入 q 退出。



vim 与 vi 有许多兼容的地方，基本命令是完全一致的。在 Ubuntu 中，实际把 vi 链接到 vim，用户使用命令 vi，实际打开的还是 vim。本节介绍的 vim 知识，其实还是基于 vi 的。但由于它的兼容性，本节学习到的内容在 vi 使用中也不会出现什么问题。

vim 有款基于 GTK 的图形界面版本，安装的方法是在终端输入“sudo apt-get install vim-gtk”，然后用命令 gvim 来启动。该软件将一些常用的功能集中在菜单和工具栏中进行操作，很大程度上方便了初学者，其界面如图 5.6 所示。

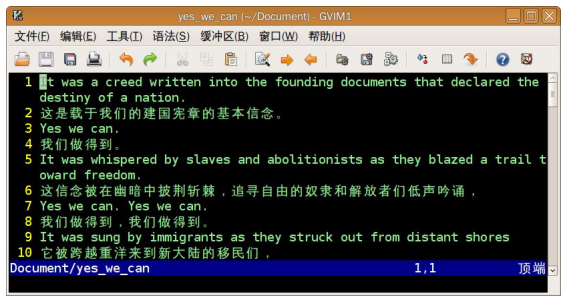


图 5.6　gvim 界面

gvim 的菜单功能很丰富，基本包括了大部分的基本操作，下面是菜单功能的简介。

- “文件”菜单除了常见的文件操作功能外，还有文件比较功能等很有特色的功能。
- “编辑”菜单中列出了可以在“.vimrc”中设置的项，如文件行号的显示，自动缩进设置、缩进设置、系统配色等功能。
- “工具”菜单主要提供了与程序编译相关的一些功能，可以设置编译程序的方法、查看编译错误输出，以及将文件以十六进制格式显示等功能。
- “语法”主要是设置语法检查、语法高亮设置等。还可以将当前已经着色的文字以彩色的方式输出到一个 html 文件中。
- “缓冲区”简单来说，相当于当前打开文件的列表。
- “窗口”提供了管理文本显示窗口的工具，可以将窗口分隔显示，以及调整窗口大小，还可以分隔出一个目录浏览窗口。
- “帮助”提供了系统帮助信息及开发者的信息等。



vim 的编写者认为，一个文件编写者在文件编写不同的时期应该处于不同的状态。如在输入文字时只关注于文字输入本身，不应该执行命令操作；而完成文字输入后就只进行相关的命令操作，不再需要输入文字。在 vim 中，也贯彻了这一思路。在编辑模式中就无法完成命令操作，而命令模式中的所有输入均被理解为命令，不会出现在文本中。在本节的阅读中，一定要勤加练习，只有将这些技巧烂熟于心，才会真正体会到 vim 的强大。

5.2.2　命令模式中的操作

命令模式可以完成简单的命令操作，一般一个字母代表一个命令。要注意的是，Linux 中命令是大小写敏感的。所以 j 与 J 代表两个完全不同的命令。j 是移动光标的指令，而 J 则是将两行合并。

1．移动光标位置

移动光标位置，一个想当然的操作就是使用方向键。但对于使用标准指法的用户来说，这就意味着手指要离开打字区，降低了工作效率。在命令模式中，移动光标的方法仍然是按键盘上某个键来实现，而且这种移动光标位置的方法在许多环境中都在使用。如在 Google Reader 中阅读 RSS 条目或者阅读 GMail 邮件时，就可以使用这些键在条目中移动。

- j：将光标向下移动一行，该键是标准指法中右手食指的位置所在。
- k：将光标向上移动一行，该键是标准指法中右手中指的位置所在。
- h：将光标向左移动一个字符。
- l：将光标向右移移动一个字符。

在 vim 中输入一些字符，然后按下 Esc 键进入命令模式，就可按这几个键在文本中移动了，感觉应该还不错。移动时有两点要注意，一是必须保证是小写状态，也就是 Caps Lock 灯没亮，二是保证处于 vim 的命令模式，如果不确定处于什么模式，可多按几次 Esc 键确保进入了命令模式。

2．删除文本

删除文本，同样由键盘中功能键来实现。如 Delete 键删除当前字符，而 Backspace 键删除当前字符的前一个字符。但键盘上的 x 键与其他功能键的实现对于高级用户来说还是更为高效的。如文件中内容如下所示，光标位于“人”字上，下面以一些实际的命令来演示 vi 的操作。

```
中华人民共和国，上海市
```

- x：删除当前字符，功能类似于 Delete 键。本例中，会删除“人”字。
- X：删除光标之前的一个字符，功能类似于 Backspace 键。本例中，会删除“华”字。
- d：删除一个字或一个词组。后面跟小写 w，删除一个字。而如果后面跟一个大写 W，就删除一个词组。
- dw：会删除“人”字及其后的内容直到第一个标点。删除后文件内容变为中华，上海市。
- dW：会删除“人”字及其后所有的文本，文件内容变成“中华”。
- dd：删除一个整行。本例中将该行完全删除，该行就没有内容了。
- D：删除到行尾，将当前光标到行尾的所有字体删除，本例中，文件内容变为“中华”。

3．添加文本

首先假设一个纯文本文件内容如下，当前的位置在第二行的第一个逗号处（也就是说，在 kis 与 serbian 的中间）：

```
中华人民共和国，上海。
kis, serbian, slovenian, estonian, esperanto, upperso
```

按下 Esc 键，回到命令模式。然后单击 i 键，进入插入状态。输入任意字符，如输入 a。就可以看到输入的字符 a 出现在光标位置前，文本成为

```
中华人民共和国，上海。
kisa, serbian, slovenian, estonian, esperanto, upperso
```

下面是一些其他的添加文本的命令。

- 添加命令 a：按下 Esc 键，回到命令模式。然后按下 a 键，就会进入添加状态。这时输入的任何字符都会添加到光标位置之后。
- 添加到行尾 A：命令状态下，按下 A 键（大写字母），会将输入的文本添加到行尾。
- 插入到行首 I：将文本添加到行首。
- 插入新行 O（大写字母）：在当前光标所在行前插入一新行。
- 添加新行 o（小写字母）：在当前光标所在行后添加一新行。

4．命令的重复执行

vim 的强大不只体现在这些快捷的命令行，vim 还可以接受将命令重复几次的命令。如“执行删除操作 5 次”，“将光标向右移动 10 次”。发出这种命令也很简单，只需要在输入前面介绍的命令前先输入数字即可。如在命令状态，输入“5x”，就将执行删除操作 5 次。

以下列的文本（行号是为了介绍方便而添加的，文本无行号）为例介绍命令重复的操作方法，设现在光标位置处于第 1 行第一个字符 I 上。

```
1   It was a creed written into the founding documents that declared the
destiny of a nation.
2   这是载于我们的建国宪章的基本信念。
3   Yes we can.
4   我们做得到。
5   It was whispered by slaves and abolitionists as they blazed a trail toward
freedom.
```

```
6    这信念被在幽暗中披荆斩棘，追寻自由的奴隶和解放者们低声吟诵，
7    Yes we can. Yes we can.
8    我们做得到，我们做得到。
```

（1）按下 Esc 键，确保处于命令模式。输入 15l（将光标向移动 15 个字符，第 3 个字符是字母 l）。

（2）输入 20x 删除后面的 20 个字符。现在第 1 行文本应该如下所示，而光标位于 n 上。

```
It was a creed nding documents that declared the destiny of a nation.
```

（3）输入 5dw，删除之后的 5 个单词。文本如下所示，光标位于 d 上。

```
It was a creed destiny of a nation.
```

（4）输入 2j，向下跳两行，然后输入 3dd，删除 3 行，现在文本成为

```
It was a creed destiny of a nation.
这是载于我们的建国宪章的基本信念。
这信念被在幽暗中披荆斩棘，追寻自由的奴隶和解放者们低声吟诵，
Yes we can. Yes we can.
我们做得到，我们做得到。
```

5．撤销操作

继续第 4 条的操作。进入命令模式后，按下 u 键会将刚才删除的两行恢复。再多按几次，就可将第 4 条最后得到的文本恢复到原始状态。

- u 撤销上一次进行的文本编辑操作。
- U 撤销对当前行所做的任何修改。
- Ctrl+R 组合键恢复上一次的操作，实际上将撤销的操作重做一次。

6．复制、粘贴

复制、粘贴在 vim 中算是比较困难的操作。因为在复制文本时又要切换到 vim 的另一种模式：可视模式。还以第 4 条中的原始文本作为编辑范本，光标仍然处于第 1 行的第 1 个字符。仍然按下 Esc 键确保处于命令模式，然后按下 Ctrl+V 组合键切换到块模式，vim 的最下面一行会出现“—可视块—”状态提示。

输入 5w 将向后移动 5 个单词，这时“It was a creed written i”被选中，按下 y 键复制选中的文本，复制后光标会回到复制前的位置。按下 P 键（大写）将复制的文本粘贴在当前光标前。现在第一行文本的内容如下：

```
It was a creed written iIt was a creed written into the founding documents
that declared the destiny of a nation.
```

vim 的可视模式有两种状态，行方式和块方式。按下 v 键进入行方式，这种方式下多行文本只能以整行的方式选中。而同时按下 Ctrl+V 组合键进入块方式，该方式多行选择时可以只选择每行的前几个字符。两者的差别可以如图 5.7 和图 5.8 所示。

图 5.7 是可视块方式，该方式可以从文本中的任意位置开始，然后“画”一个矩形，该矩形内的文本均会被选中。然后可以执行相应的复制或剪切、删除命令。

而如图 5.8 所示的可视方式，将文本的一行作为一个整体看待。如果要选中该行，则须将该行所有内容均选中。不能像可视块方式那样，只选中所有行的前 10 个字符。

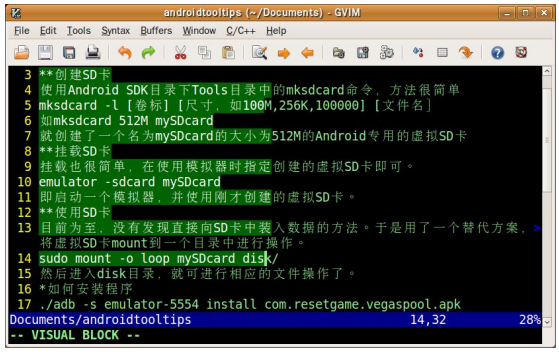


图 5.7　可视块方式

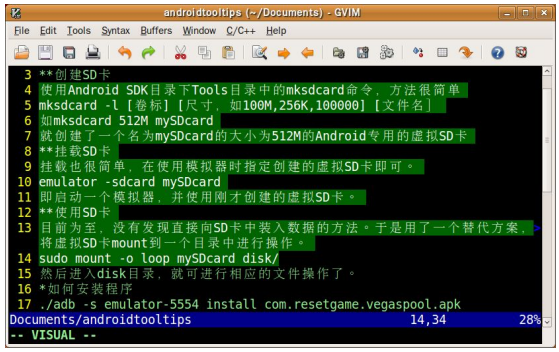


图 5.8　可视模式

总结复制、粘贴的命令。

- Y：yank，复制当前行的内容。
- y：复制当前选中的内容。
- P：（大写）将复制的内容粘贴到当前光标位置之前。
- p：（小写）将复制的内容粘贴到当前光标位置之后。

需要注意的是，前面的复制命令复制的内容只能在 vim 中使用，如果要将 vim 中的文本内容复制到系统剪贴板，以便其他程序使用，就须使用“"+y”命令。

7．替换

本例以下列文本内容为例介绍替换命令，也就是说所有的操作均作用于该行，当前的光标位于行首。

> 这信念被在幽暗中披荆斩棘，追寻自由的奴隶和解放者们低声吟诵，

- r：设光标位于行首，按下 r 键进入单字替换状态，输入任意字符，如“那”，则该行文字成为：“那信念被在幽暗中披荆斩棘，追寻自由的奴隶和解放者们低声吟诵，”。然后 vim 会回到命令模式。等待下一条命令。
- R：按下大写 R 键，进入行替换状态，现在输入“那”后，光标移动到下一个字条，等待继续输入文字来替换光标所在位置的字符。

8．单词替换命令

前面介绍的 R 键替换方式不太灵活。如果只是想要将一个单词中的某几个字符替换，而且

字符数量不相等时，如要把 ruby 修改为 rubber。这样的情况下，R 键就显得不够用了。所以，vim 提供了单词替换命令 c。如本例，将光标位于 u 处，再输入 cw 进入单词替换功能，就可以输入 u 后面的字符 bber 完成替换。

9．行跳转

要将光标置于某行，最简单的方法是输入“行号＋G”，如要将行跳转到 15 行，直接按 15G 就可以看到光标已经位于 15 行了。直接按“G”键会将光标置于最后一行。

10．文本查找

最简单实用的查找文本的方法是将光标置于要查找的文本之上，然后按下“*”（同时按下 Shift+8）键，就可以在文本中搜索光标处的单词。英文中是以空格分界，对于中文而言，是以标点符号分界，两个标点之间的所有中文文字都会当做一个单词来查找。

- #：查找当前单词。与“*”功能类似，不过是从当前位置向文件头搜索，方向与“*”相反。这是两个很重要的技巧，长期在 vim 网站的技巧排名中名列第一。
- %：查找与当前光标处括号配对的另一个括号，可以包括三种括号()、[]、{}。
- /：进入文本查找状态，窗口底端会出现搜索框，在其中输入要查找的字符，然后按 Enter 键。如果找到了输入的字符，会将停留于该字符所在的位置，如果没找到，会在窗口底端给出一个错误提示。
- ？：进入反向查找状态。与“/”操作方式类似，不过搜索方向相反。
- n：如果执行“/”，“？”进行查找，就可以按 n 键重复这个查找过程。

5.2.3　命令行模式可执行的命令

在命令模式中，按下“:”键就进入了命令行模式。光标会出现在屏幕底端，等待输入命令。

命令行模式中可以使用的命令要远多于命令模式中键盘的快捷键，一个易见的理由是键盘的键的个数是有限的。而在命令行模式中，可以输入一个字符串作为命令，想象一下，字母可以有多少种组合方式。

1．文件操作

文件的打开，保存都可以在命令行中来实现。本节介绍一些与文件操作相关的命令，如打开文件，保存文件，插入文件等。

- 打开文件：edit 或简写为 e。在命令行中输入“e ～/.vimrc”可打开用户主目录中的.vimrc 文件。
- 保存文件：write 或简写为 w。将当前正在编辑的文本内容保存在文件中。如果当前编辑的文本还没有保存到一个文件中，后面还需加上文件名，如“w ～/myvim”，否则 vim 会提示没有文件名，无法完成保存。如果其是作为一个文件被打开或者已经保存为一个文件，就可直接用 w 保存。保存文件命令还可保存文件的部分内容，如“12,26w smple”，将当前文本中的第 12 到 26 行内容保存到 simple 文件中。
- 另存为：sav。如果想将当前打开的文件以另一个文件名保存，需用 sav 命令将其另存。
- 插入另一个文件：r。如果要在当前文件中插入另一个名为 test 的文件，就可以用 r test 将 test 文件中的内容插入。

2．文本替换命令

文本替换也是在实际操作中很常见的一种操作。vi 提供了强大的文本替换功能，可以指定

替换的范围（如整篇文章还是特定的范围），也可以使用简单的正则表达式来指定查找的内容等。下面通过一个例子来演示其功能。如“12,36s/windows/linux/g”命令可以完成将第 12 行 36 行中所有的 Windows 替换为 Linux 的功能。下面是该命令的详细解释。

- 12,36 指定了文本替换的范围，可以指定为“%”，意思是当前文件中所有的文本，如不指定则只替换当前行。
- s 是命令的关键，指定命令执行的是替换操作。
- /windows 指定搜索内容为 windows，可以使用正则表达式来指定搜索的内容。
- /linux 指定将搜索内容替换为 linux。
- g 指定替换的方式为全局替换，意思是将所有匹配的内容全部替换，如果不指定则只替换一次。

3．执行外部命令

在命令行中还可执行 Linux 的相关命令。方法是在命令前加“!”。如要显示当前目录中的文件目录信息，可以用“!ls”来显示。使用外部命令，可以完成许多复杂的工作。如要将当前文本中第 2 行到第 100 行的内容按第 3 列排序，可以使用命令“2,100!sort”。文本中的内容就会成为一个有序的排列。“%nl”则是给当前所有文本行加上编号。

4．强制命令

如果需要忽略 vim 的警告，强制使用某些命令，可以在其后加上“!”。如对于某文件进行了编辑，而没有将修改保存，这时执行退出命令 q 就会得到一个警告信息，而无法退出。但如果给命令加上“!”使命令强制执行，就会忽略警告，直接退出。

5．获得帮助

vim 提供了完整的帮助信息，对于每条命令均可以通过“help”获得帮助信息。如“help edit”会得到打开文件命令的详细帮助。vim 的手册是 help user-manual。

5.2.4 vim 的基本设置

vim 中有许多功能，默认情况下这些选项是关闭的，为充分发挥其能力，可以使用配置文件将这些选项打开。而且 vim 的配置文件设置也是一项很复杂且耗时长的工作。很可能需要在工作、生活过程中不断对其进行完善。

1．建立配置文件

对于用户来说，vim 的配置文件是位于主目录.vimrc 文件的。如果系统中没有该文件，可以从系统的一个示例配置为基础，建立自己的配置文件。系统的示例配置文件位于：/usr/share/vim/vim71/vimrc_example.vim。为了将刚才学习到的知识结合起来，就用 vim 编辑这个配置文件：

（1）使用“vim ～/.vimrc”命令打开 vim，这样打开如果该文件存在，则显示该文件的内容，如果不存在，则新建了该文件。

（2）进入 vim 的命令模式，在命令行中输入“r /usr/share/vim/vim71/vimrc_example.vim”将系统示例配置插入到文件中。

（3）在命令行中输入“:wq”，保存退出。

（4）再使用“vim ～/.vimrc”命令打开 vim，就可以看到现在这些配置文件已经有了漂亮的颜色。

2．设置配置文件

不过，总体而言，系统的示例配置功能还是不能让人满意的。所以还需要再增加些配置项，以使 vim 使用更为顺畅。下面是一个配置文件，读者可以根据自己的需要灵活设置各选项。前面有“"”符号的是注释文件，可以在不需要的配置项前加上“"”来注释这些内容。

```
~/.vimrc
"vim 的配置文件
"自动检测文件编码，设置就可以正确打开绝大多数中文编码的文件，不会出现乱码
set fileencodings=utf-8,gbk,ucs-bom,cp936,big5,utf-16,euc-jp
"打开行号显示
set nu!
"配色方案设置为黑底绿字
color murphy
"使用非兼容 vi 的模式，如果使用 vi 兼容模式，会无法使用方向键，没有多次撤销功能
set nocompatible
"将 vim 历史记录的行数设定为 50
set history=50
" 在处理未保存或只读文件的时候，弹出确认
set confirm
" 与系统其他程序共享剪贴板
set clipboard+=unnamed
" 自动侦测打开或新建的文件类型
filetype on
" 载入文件类型插件，可以执行相应的操作。如 LaTeX 插件检测到打开或新建的文件是 TeX 文件时，会自动启动 LaTeX 菜单。
filetype plugin on
" 为特定文件类型载入相关缩进文件
filetype indent on
" 打开语法高亮
syntax on
" 状态行颜色
highlight StatusLine guifg=SlateBlue guibg=Yellow
highlight StatusLineNC guifg=Gray guibg=White
" 禁止 vim 自动备份文件
set nobackup
" 不要生成 swap 文件，当 buffer 被丢弃的时候隐藏它
setlocal noswapfile
set bufhidden=hide
" 增强模式中的命令行自动完成操作
set wildmenu
" 在状态行上显示光标所在位置的行号和列号
set ruler
" 使退格键(Backspace)正常处理 indent、eol、start 等
set backspace=2
" 允许 backspace 和光标键跨越行边界
```

```
set whichwrap+=<,>,h,l
" 可以在 buffer 的任何地方使用鼠标(类似 office 中在工作区双击鼠标定位)
set mouse=a
set selection=exclusive
set selectmode=mouse,key
" 启动的时候不显示系统提示信息
set shortmess=atI
" 通过使用: commands 命令，告诉我们文件的哪一行被改变过
set report=0
" 不让 vim 发出讨厌的嘀嘀声
set noerrorbells
" 高亮显示匹配的括号
set showmatch
" 匹配括号高亮的时间(单位是 1/10 秒)
set matchtime=5
" 在搜索的时候忽略大小写
set ignorecase
" 不要高亮被搜索的句子(phrases)
set nohlsearch
" 打开递增搜索模式，就是输入一个字符就开始搜索
set incsearch
" 禁止光标闪烁
set novisualbell
" 总是显示状态行
set laststatus=2
" 打开自动缩进功能
set autoindent
" 为 C 程序提供自动缩进
set smartindent
" 使用 C 样式的缩进
set cindent
" 制表符宽度为 4
set tabstop=4
" 统一缩进为 4
set softtabstop=4
set shiftwidth=4
" 不自动换行，这在处理格式严格的文本中比较有用，如果不换行，在阅读一些纯文本无格式文档时会造成很多不便
set nowrap
```

5.2.5 安装插件概述

vim 的功能虽然强大，但是总有些用户仍不满足。所以在 vim 中仍然有许多强大的插件来扩充其功能。在介绍这些插件前，先看一个 vim 配置了某些插件后的界面图，如图 5.9 所示。

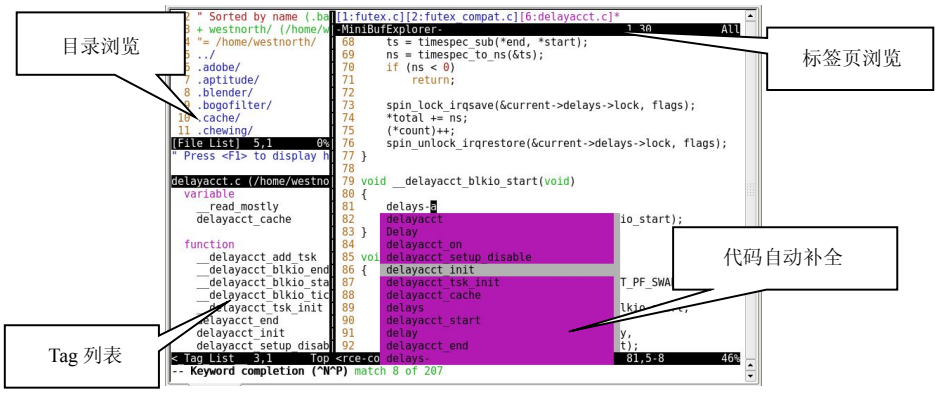


图 5.9　vim 界面图

下面几节将讲解几种常见的插件的使用。

5.2.6　目录浏览器

vim 从 7.1 开始，支持目录浏览功能；也就是说，可以把 vim 当文件管理器来使用。这个是系统默认就安装的插件，可以直接使用。打开 vim，然后在命令行输入“e /bin”命令，显示如图 5.10 所示。很明显，这是个打开目录的命令，参数中只给出了目录名，没有给出文件名。但即使这样，vim 也不会报错，而是将目录打开以供浏览。

从如图 5.9 所示中可以看到，当前目录是“/bin”，当前目录显示排序方式是以文件名排序，“-”是到上级目录，D 是删除文件，R 是重命名文件，s 可以改变排序方式，x 可以运行文件。

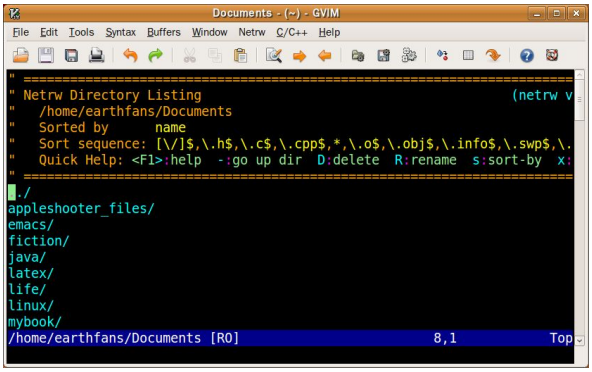


图 5.10　vim 打开目录

5.2.7　Tag 列表

vim 可以很好地支持 Tag 文件。所谓 Tag 文件，就是一些记录了文件中函数名、变量、结构等信息的一个列表，Tag 文件还包括了这些信息所在的文件目录及在文件中的行数。如下面的文本来自一个 Tag 文件：

```
CSIZE    include/termios.h    203;"    d
```

这行文本的意思是 Tag 的名字叫 CSIZE，位于当前目录子目录 include 中文件 termios.h 的

第 203 行。虽然 Tag 文件很清楚地列出了这些变量的位置，但如果不能有一个方便的方式查看，也没有什么价值。当然 vim 提供了许多利用 Tag 文件来提高效率的方法。

首先要介绍浏览 tag 文件。这也是通过一个插件来实现的。该插件名为 Taglist，可在 http://vim.sourceforge.net/scripts/script.php?script_id=273 下载。现在最新版本是 4.5 版。

将下载的文件解压，并将插件 plugin 目录中的 taglist.vim 文件复制到“～/.vim/plugin”目录。如果需要，还可以将插件 doc 目录中的 taglist.txt 文件复制到“～/.vim/doc”目录中，插件就安装好了。插件安装一般均可采用这种方式，将 vim 文件复制到“～/.vim/plugin”目录，将帮助文件复制到“～/.vim/doc”目录，就安装成功了。安装后，还需要在 vim 的配置文件中加上两句，内容是

```
let Tlist_Show_One_File=1
let Tlist_Exit_OnlyWindow=1
```

安装了 taglist 后，就可以在 vim 中使用 Tlist 命令打开 taglist 列表了。而且还多了一项很重要的功能：代码的自动补全功能。使用代码自动补全的方法如下。

（1）在 vim 的命令行中运行“!ctags -R .”，注意最后的点。目的是建立 Tag 文件。

（2）在输入代码的中途，按“Ctrl+P”或“Ctrl+X Ctrl+O”组合键就会按照 Tag 文件中的变量等信息提供自动补全的提示。

5.2.8 标签页浏览

如果打开的文件多于一个，会发现要打开其他文件很不方便。这时就需要这个 minibufexpl 插件。该插件可以在窗口顶端显示文件名列表，以便用户切换。安装了该插件的 vim 界面如图 5.11 所示。

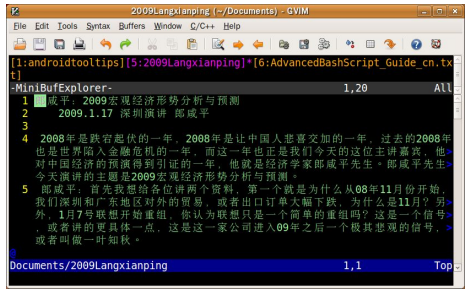


图 5.11　标签页浏览

可以看到该插件给每个文件均编了号，如“1:androidtooltips”。这样在命令行中输入“b1”就会打开该文件。在图形模式中，也可以使用鼠标双击文件名打开该文件。该插件可在 http://www.vim.org/scripts/script.php?script_id=159 下载，得到的是扩展名为 vim 的插件文件，将该文件放入“～/.vim/plugin/”目录即完成安装。

5.3 强大的 Emacs

Emacs 号称是 Linux 系统中最为强大的文本编辑器。而且称其为文本编辑器还远不能概括其功能。除了强大的文本编辑和编程辅助功能外，还可以收发邮件、看新闻组、浏览网页，最

让人惊奇的是其还可以听音乐、查词典等。最新版的 Emacs 还可以看图片、阅读 PDF 等。所以有人戏称：“世界上有三种程序员，一流的程序员用 Emacs，二流的程序员用 vim，其余的程序员用其他”。虽然有点夸张，但准确地说，在许多人心目中，Emacs 已经成为一种信仰、一种文化。本节就简单地介绍 Emacs 的基本功能。

5.3.1　基本配置

在 Ubuntu 中，安装 Emacs 是很容易的，其软件源文件中有两种版本的 Emacs：一种是当前的稳定发布版，可以在终端输入“sudo apt-get install emacs22-gtk”，即可完成安装；另一种则是提前发布的预览版，要尝试新功能的用户可以安装这个版本，方法是运行命令“sudo apt-get install Emacs-snapshot-gtk”进行安装。



预览版没有经过充分的测试，性能可能并不稳定，不能在极为重要的场合使用，但却包含了一些新功能，比较适合于追求新功能的用户使用。

稳定版运行的方法是直接在终端输入“emacs”命令或在主菜单“应用程序”|“附件”中打开 Emacs 22 GTK 即可，这是图形界面版本。而要运行字符界面版，可以选择 Emacs 22 client。而预览版与稳定版在同一个菜单中，很容易可以看到。由于 Ubuntu 软件源中的 Emacs 预览版支持 xft 字体，以及文档视图等高级特性，本节以该版本为例进行介绍。

同 Linux 系统中的大多数软件一样，默认情况下 Emacs 性能没有完全发挥，需要进行初步的配置。首先要做的就是编写配置文件。Emacs 的主要配置文件是“～/.emacs”，如下所示是 Emacs 官方推荐的配置，将这些内容加入配置文件中，就可完成 Emacs 的初步配置。

```
(set-language-environment "Chinese-GB18030")    ;设置中文环境支持
(setq default-major-mode 'text-mode)            ;启动时默认进入 text 模式
(setq inhibit-startup-message t)                ;禁用启动画面
(global-font-lock-mode t)                       ;语法高亮
(auto-image-file-mode t)          ;打开图片显示功能，这是个很重要的功能，必须开启
(fset 'yes-or-no-p 'y-or-n-p)     ;以 y/n 代表 yes/no，让输入简洁化
(column-number-mode t)            ;显示列号
(show-paren-mode t)               ;显示括号匹配
(display-time-mode 1)             ;显示时间，以下两行列出的是显示的格式
(setq display-time-24hr-format t)
(setq display-time-day-and-date t)
(tool-bar-mode nil)               ;去掉工具栏
(scroll-bar-mode nil)             ;去掉滚动条，台式机用户专用，笔记本用户应将其注释掉
(setq mouse-yank-at-point t)      ;支持中键粘贴
(transient-mark-mode t)
(setq x-select-enable-clipboard t)              ;支持 emacs 和外部程序的粘贴
(setq frame-title-format "westnorth@%b")        ;在标题栏提示目前所在位置
(setq default-fill-column 80)                   ;文本显示的宽度设为 80 列
(require 'ido)
   (ido-mode t)                   ;在用“C-x C-c”打开文件时在状态栏显示提示信息
```

设置了这些选项后，启动 Emacs 的 GTK 版本，就可以看到 Emacs 已经发生了些许变化。

但现在还有一个问题，就是中文字体的显示，默认显示的中文字体看起来很不舒服，需要再配置中文字体。字体配置可在.Xresources 中设定。将如下的代码放进该文件，然后使用命令“xrdb -load ～/.Xresources”刷新 X 设置数据库。

```
Xft.antialias: 1
Xft.hinting: 1
Xft.hintstyle: hintfull
Emacs.FontBackend: xft
```

再在～/.emacs 中加入如下内容：

```
(set-default-font "Bitstream Vera Sans Mono-12")
(set-fontset-font (frame-parameter nil 'font)
            'han '("WenQuanYi zenhei" . "unicode-bmp"))
; Emacs.Font:fontset-chinese
```

现在 Emacs 启动后，就会使用自己满意的字体显示文本内容了。

5.3.2 设置系统配色方案

Emacs 的强大在很大程度上要归功于插件的丰富。Emacs 插件的数量非常巨大，可以满足不同的需求。本节介绍一些对初学者来说比较实用的插件。

有个小插件，名字是 color-theme.el。专门用来设置 Emacs 的配色，其中包含了 70 多种颜色样式。可以在这个地址得到该插件 http://linux.duke.edu/～mark/emacs/color-theme.el。下载回来后将其放在一个目录中。建议给 Emacs 的插件专门建立一个目录，如“～/.Emacs.d”。然后将插件放进该目录，将如下代码加入到.Emacs 中：

```
(load-file ".emacs.d/color-theme.el")
(color-theme-goldenrod)          ;选择自己的配色方案，这是黑色背景，金色字体的方案
```

经过这些设置后，就可以在 Emacs 菜单 Tools|Color Themes 中浏览所有的颜色方案。然后使用自己喜欢的方案名将上面配色方案的名字替换，重启 Emacs 就可以看到新的颜色效果。

5.3.3 行号显示

vim 中很容易打开行号显示，但在 Emacs22 中（在 Emacs23-snapshot 版本中，可以直接使用命令 linum-mode 打开行号显示），不得不安装插件来实现。推荐一款国人写的非常优秀的插件，在这儿下载该插件 http://ann77.stu.cdut.edu.cn/elisp/display-line-number.el。得到一个插件文件，将其保存到“～/emacs.d/目录”，然后在“～/.emacs”中加入如下代码。

```
 (load-file ".emacs.d/display-line-number.el" )     ;加载显示行号模块
(global-display-line-number-mode 1)                 ;如果想所有打开的文件都显
示行的话就打开下面的注释
(setq display-line-number-format "%4d|")            ;设置显示格式
;下面几行是操作说明，包括如何打开显示行号模式等
;M-x display-line-number-mode                       ;使用方法是按“M-x”
;M-x display-line-number-mode-on                    ;启动显示行号的模式的命令
;M-x display-line-number-mode-off                   ;关闭显示行号的模式的命令
```

```
;(add-hook 'c-mode-hook 'display-line-number-mode) ;仅对某种 mode 启动显示行号的模式
```

5.3.4　标签页浏览

使用 Emacs，不可能每次只打开一个文件。但要多个文件同时打开，寻找特定的文件就比较困难。当然可以按“C-x X-b”来浏览，但毕竟很不直观。要实现标签页浏览功能，就要使用 tabbar 插件。在这里下载 http://www-cdf.fnal.gov/～sthrlnd/emacs/tabbar.el。将下载的 el 文件保存到～/.emacs.d 目录，然后在～/.Emacs 中加入如下代码：

```
;;标签浏览功能配置，使用 tabbar
(load-file ".emacs.d/tabbar.el")
(require 'tabbar)
(tabbar-mode)
(global-set-key (kbd "") 'tabbar-backward-group)
(global-set-key (kbd "") 'tabbar-forward-group)
(global-set-key (kbd "C-`") 'tabbar-backward)
(global-set-key (kbd "C-<tab>") 'tabbar-forward)
;设置 tabbar 字体
(set-face-attribute 'tabbar-default-face
        nil :family "Tahoma")
```

5.3.5　建立 C 语言开发 IDE

建立开发使用的 IDE，可以使用 cedet、ecb 的插件。在 Ubuntu 系统中安装很简单，在终端使用命令“sudo apt-get install ecb cedet-contrib”，即可完成安装。安装后，将如下代码加入～/.emacs 中：

```
;;cedet 的设置
(add-hook 'texinfo-mode-hook (lambda () (require 'sb-texinfo)))
    (semantic-load-enable-code-helpers)
(require 'ecb)              ;本行开始是对 ecb 的设置
(custom-set-variables
 '(ecb-options-version "2.32"))
(require 'ecb)
(setq ecb-auto-activate t   ;默认打开 ecb，如果不是工作时使用的话，最好取掉该项
           ecb-tip-of-the-day nil
           ecb-tree-indent 4
           ecb-windows-height 0.5
           ecb-windows-width 0.20
           ecb-auto-compatibility-check nil
           ecb-version-check nil
inhibit-startup-message t)
```

启动 Emacs，按下 Alt+X 组合键，输入“sustomize-option”，会提示输入变量，输入“ecb-source-path”开始对 ecb 进行设置。设置完成后，重新启动 Emacs，打开一个 C 语言文件，

就可以看到多了许多菜单与窗口，如图 5.12 所示。

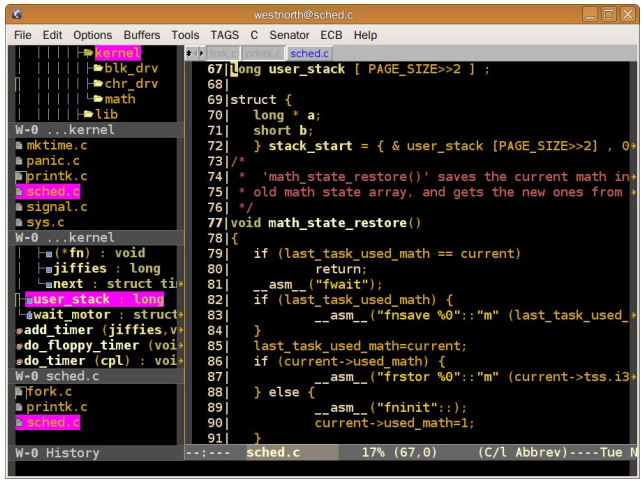


图 5.12　用 Emacs 建立的开发环境

5.3.6　多内码支持

不可讳言，Emacs22 对于多编码的文件支持并不好，经常出现打开文件乱码的情况。如将编码设置为固定的某个编码后，打开另外编码的文件就会出现乱码。unicad 就是专门用来解决这个问题的插件，下载地址是 http://cid-3643184002e3d5e6.skydrive.live.com/self.aspx/Public/unicad.zip。将下载的文件解压后，将其中的 unicad.el unicad.el 移动到～/.emacs.d 目录，然后在～/.Emacs 里添加一句如下语句：

```
(load-file ".Emacs.d/unicad.el")
```

这样，无论打开什么编码的文件，Emacs 均会调用该插件来自动判断编码，正确打开。在 Emacs23 中不用插件也可以正确打开各种编码的文件。



重新设置了配置文件后，可以重启 Emacs 以使其生效。也可以在 Emacs 中打开 ~ /.Emacs 文件，然后执行命令“M-x eval-buffer”，便可以使配置文件立即生效。

5.3.7　基本操作

Emacs 的功能强大，但相对来说，入门的难度甚至要大于 vim。不过，Emacs 有个好处，在文本编辑时，从某种意义上来说，只有编辑模式与命令行模式，而不像 vim 有四种模式。处理文本的相关命令均是通过组合键来实现的。但其命令实在比较复杂，像打开文件、保存文件这样常用的功能都需要按四个键才能实现。



本节对组合键的描述遵循 Emacs 的惯例。例如“C-”表示按下 Ctrl 键，而“M-”表示按下 Alt 键。“S-”代表按 Shift 键，“s-”在 Linux 下对应 WIN 键（就是那个有微软的视窗图标的键，一般在左边的 Ctrl 键与 Alt 键之间），“RET”代表 Enter 键。

1．操作前的预备知识

在实际使用 Emacs 来完成各种复杂的工作前，先要了解其基本知识。这里仅介绍 Emacs 的命令行模式与命令重复的方法。

命令行模式，同时按下“M-x”键即进入命令行模式。光标停留在窗口底端，可以输入相关的命令。Emacs 的命令同样种类繁多，数量巨大。在输入命令的过程中，同样可以使用 Tab 键进行自动补全，如果该命令进行了键绑定，则在输入命令并按下 Enter 键执行后，会提示绑定的键。

命令重复，在 Emacs 中也可以让命令重复执行，不过方法比 vim 稍微复杂一些。一种方法是按住 Alt 键，然后输入数字（如 8），再在命令行输入某个命令，就会将该命令执行 8 次。第二种方法是在命令行输入“C-u”，然后输入数字，再输入命令，这样也可以重复执行命令。

2．移动光标

在文字中移动是文本编辑器使用的入门课程，vim 中是这样，Emacs 仍然是这样。用上文所述知识打开一个文件开始练习移动光标位置的方法。移动光标的方法如下所示。

- C-v：向下翻一页。
- M-v：向上翻一页。
- C-l：刷新屏幕，并将光标置于屏幕中央。
- C-p：向上移动一行。p 的意思是 previous。
- C-b：向左移动一个字符。b 代表 backward。
- M-b：向左移动一个单词。
- C-f：向右移动一个字符。f 代表 forward。
- M-f：向右移动一个单词。
- C-n：向下移动一行。n 代表 next。
- C-a：将光标移到当前行首。
- M-a：将光标移到光标所在句子的句首。句子以句号作为标志。
- C-e：将光标移到行尾。
- M-e：将光标移到句尾。
- M-<：将光标移动到文章的开头。
- M->：将光标移动到文章的末尾。

3．分隔窗口

Emacs 可以将窗口分隔为几部分，各部分可以显示不同的文件，也可以显示同一个文件，这在编辑一个很长的文件时很有用。将窗口分隔的命令很简单。

- C-x 2：将当前窗口水平分为两个。
- C-x 3：将当前窗口垂直分为两个。
- C-x 1：将除了当前窗口外的其他窗口全部关闭。
- C-x o：切换到另一个窗口中。
- C-x 4 C-f：在新的窗口中打开文件。
- C-M-v：滚动另外一个窗口的内容。
- C-x 0：关闭当前窗口。

4．文字的删除操作

删除操作，除了使用键盘上常用的几个键外，还可以使用下面的组合键，而且使用这些快

捷键会更为快捷与灵活。

- M-w：复制选中区域。
- C-d：删除光标后的一个字符。
- M-<backspace>：移除光标前的一个词。
- M-d：移除光标后的一个词。
- C-k：移除从光标到“行尾”间的字符。
- M-k：移除从光标到“句尾”间的字符。
- C-w：移除选中的内容，相当于剪切。
- C-y：将最后一次移除的内容粘贴到光标处。
- M-y：执行 C-y 后，就可以执行这个命令将前几次删除的内容粘贴在光标处。
- C-x u：撤销最后一次文字编辑所做的修改。
- C-x r k：剪切选中区域。
- C-x r y：粘贴上一次剪切的区域。
- C-x r t：将指定字符插入到选中区域的行首。

5．缓冲区（BUFFER）

在 Emacs 中，每打开的一个文件被称为一个缓冲区。Emacs 可以拥有多个缓冲区，下面介绍的命令用来管理这些缓冲区。

- C-x C-b：列出当前所有打开的缓冲区（包括打开的文件、缓冲区信息列表等）。
- C-x b：后面跟上名称，就可以切换到相应的缓冲区。
- C-x s：保存所有缓冲区的内容，如果缓冲区有多个打开的文件，用该命令可将保存对所有的缓冲区所做的修改。
- M-x auto fill mode：打开（或关闭）字绕回模式。

6．搜索

在文本编辑中，文本内容的搜索是很重要的内容。而 Emacs 提供了强大的搜索功能，可以完成复杂的搜索工作，下面是这些搜索命令的介绍。

- C-s：正向搜索，意思是从当前位置向文件尾部依次搜索。Emacs 是输入一个字符就开始搜索，然后随着输入内容的增多而逐渐缩小匹配范围，这种搜索方式叫做递增搜索。Emacs 会将所有匹配的内容高亮显示。如果要切换到下一个匹配的项目上，可以一直按 C-s。
- C-r：反向搜索。
- C-M s：带正则表达式的增量搜索。
- M-%：替换前提示。
- C-M-%：带正则表达式的替换。

7．文本换位

在文本输入过程中，可能会出现将字母的位置输入错误，或者出现需要调整文本行的位置的情况，这时就是文本换位命令的用武之地了。下面是这些命令的简介。

- C-t：将当前字符和前一个字符互换位置。
- M-t M-x：将当前单词与前一个单词换位。
- C-x C-t：将当前行与前一行互换位置。

8．书签

如果经常需要打开某个文件或某个文件的某个位置，可以使用书签功能来提高效率。

- C-x r m <name>：设置书签，<name>可以是一个字串。
- C-x r b <name>：跳转到书签。
- C-x r l：书签列表。
- M-x bookmark-delete：删除书签。
- M-x bookmark-load：读取存储书签文件。

9．获得系统帮助

由于 Emacs 的命令数量极为繁多，没有人可以全部记住。所以，提供强大的用户帮助是一件极其重要的工作。而且，Emacs 的帮助使用也比较特别，所以下面列出了一些引起系统帮助的使用方法。

- C-h：在后面输入一个字符以说明需要哪些方面的帮助信息。如跟一个 a，就可以得到与输入字符匹配的命令的信息。b 是键盘绑定信息。跟一个？可以得到其可提供的帮助信息的列表。
- C-h c：获得组合键的帮助信息。想得到更多的信息，请把 C-h c 换成 C-h k 试试看。
- C-h k：获得比 C-h c 更详细的帮助信息。
- C-h f：获得一个详细的函数相关信息。

10．文件操作

文件操作在任何文字编辑软件中均是重要的功能，下面列出了文件操作的几项重要内容。

- C-x C-f：在窗口底部出现文件名称输入区，如果依照前文所述进行了相关设置，还会进行智能提示。输入.vimrc 就会打开 vim 的设置文件。
- C-x C-s：保存当前操作的文件。
- C-x C-c：退出 Emacs。

5.4　正则表达式简述

简单地说，正则表达式是一种文本匹配的模式，而且其主要匹配高度格式化的文本。如找出“这个日志中所有人的电子邮件地址”、“将数据文件中日期早于 2008 年的记录挑选出来”等。这个功能在处理数据文件和查找、替换文本方面有着极其重要的作用。对于前面所述的文本编辑器，如果缺少了正则表达式的支持，可想而知其效率是会降低几个数量级的。而第 6 章介绍的文本处理命令如 grep、awk、sed 更是大量使用了正则表达式。本节介绍编写正则表达式的常识。

正则表达式就是类似于“\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*”这样的匹配模式，不过不要被这些难以读懂的文本吓倒，理解了其基本原理后，正则表达式并不是很难的。

1．元字符

所谓元字符，就是正则表达式中最基本的字符。正则表达式中的元字符包括\、*、?、{、}、[、]、(、)、+、.、^、$。这些字符在正则表达式中着特殊的意义。这些字符也是正则表达式的核心，认识了元字符，就可以读懂简单的正则表达式了。由于正则表达式是用来匹配文本的，

所以将下面的内容作为匹配的源文件，本节的例子，均是以下文为基础的。

```
    it was a creed written into the founding documents that declared the destiny
of a nation.
    这是载于我们的建国宪章的基本信念。
    Yes we can.
    我们做得到。
    It was whispered by slaves and abolitionists as they blazed a trail toward
freedom.
    这信念被在幽暗中披荆斩棘，追寻自由的奴隶和解放者们低声吟诵。
    Yes we can. Yes we can.
```

首先来看这个\，其是正则表达式系统中重要的转义字符。后面跟不同的字符就有不同的含义。上文列出的元字符，如果在正则表达式中直接使用代表的就不是其自身的那个字符，而是某个特殊的意义。如*表示任意一个字符，而?表示一个字符。如果要将这些元字符作为文本内容来查找，就必须在其前加\，如要寻找*，就需要使用*来进行匹配。下面列出一些正则表达式中的以\开头的元字符。

- \b：匹配一个单词的开头或结尾，但并不特别匹配某个字符，只是匹配单词前或后的搁置。对于中文来说，单词的概念成为两个标点之间的句子。如“\b 我们\b”只匹配第二行句号后的“我们”，而不匹配“我们做得到”。
- \d：匹配一个数字。如“\d\d\d”会匹配 333、284 等 3 位数。
- \s：匹配任意空白符号，如空格、Tab 等。
- \w：匹配所有字母、数字、下画线，也匹配一个汉字。
- \W：匹配不是“\w”的字符，如字母、数字、下画线、汉字等字符。
- \S：匹配任意不是空白符的字符。
- \D：匹配任意非数字的字符。
- \B：匹配不是单词开头或结束的位置。
- \f：匹配一个换页符。
- \n：匹配一个换行符。
- \r：匹配一个回车符。
- \d+：匹配一个或更多连续的数字。这里的+是和*类似的元字符，不同的是*匹配重复任意次（可能是 0 次），而+则匹配重复一次或更多次。

2．简单的语法

正则表达式中除了跟在“\”后的元字符外，还有一些特定的符号，这些符号可以指定匹配的模式。掌握了元字符与这些语法后，就可以读懂大多数正则表达式了。下面列出了部分正则表达式语法，以供查阅。

- .：匹配除换行符以外的任意字符。
- ^：匹配字符串的开始。
- $：匹配字符串的结束。
- ：重复零次或更多次。如 abc*匹配 ab、abccc 但不匹配 ac。
- +：重复一次或更多次。如 abc?匹配 abc、abccc 但不匹配 ab。
- ?：重复零次或一次。如 abc?匹配 ab、abc 但不匹配 abb。

- {n}：重复 n 次。如“\d[3]”匹配 3 位数字。
- {n,}：重复 n 次或更多次。如“\d[3,]”匹配 3 位或 3 位以上的数字。
- {n,m}：重复 n 到 m 次。如“\d[3，8]”匹配 3 位到 8 位的数字。
- []：可选项。如[abc]匹配 a 或 b 或 c。
- [^3]：“^”符号是排除符，本例匹配除了 3 以外的任意字符。
- [0-9a-f]：还可以括号内列范围，本例匹配所有数字及任意从 a 到 f 的字母。
- [^b8]：还可以排除一个列表中的内容，如本例匹配除了 b、8 以外的任意字符。

3．简单的例子

这里有几个简单的例子，可以参照前文的说明来理解正则表达式的意义。

- (^\s*)|(\s*$)：匹配句首、尾空格。“|”表示逻辑“或”。
- ^\d+$：匹配正整数+0。
- ^[0-9]*[1-9][0-9]*$：匹配正整数。
- ^(-?\d+)(\.\d+)?$：匹配浮点数。
- ^[A-Za-z]+$：匹配由所有英文字母组成的字符串（包括大、小写）。
- ^[A-Za-z0-9_]+$：匹配由数字和 26 个英文字母与下画线组成的字符串，这在英文世界里与\w 有相同的意义。
- http://([\w-]+\.)+[\w-]+(/[\w- ./?%&=]*)?：匹配网址 URL。
- ^[\w-]+(\.[\w-]+)*@[\w-]+(\.[\w-]+)+$：匹配 E-mail 地址。

4．逻辑“或”

有时，可能需要匹配多种模式，如电话号码，分为 3 位区号，8 位号码，或者 4 位区号，7 位本地号码这两种。如果用一个正则表达式来匹配这两种可能存在的格式，就用“|”。

0\d{2}-\d{8}|0\d{3}-\d{7}这个表达式能匹配两种上文所述格式的以连字号分隔的电话号码。其中“0\d{2}－\d{8}”匹配 3 位区号，8 位本地号码的电话。而“0\d{3}-\d{7}”匹配 4 位区号，7 位本地号码的电话，中间用“|”分隔，表示匹配该符号前后两种模式的任一种。

5．分组

如果需要匹配字符串组合的重复形式时就需要使用分组功能。分组功能是用小括号“(”、“)”将需要的字符串括起来，则被括起来的字符串称为一组。如“(\d{1，3}\.){3}\d{1，3}”这个正则表达式就是一个分组的例子。

先看括号里的内容“(\d{1，3}\.)”，很容易地分析出该内容匹配一个 1 到 3 位数后面跟一个“.”的文本，如“388.”。其后紧跟了“{3}”，说明将前面的内容重复 3 次。前面的内容是包含于一个括号中的，也就是说后面的操作符应该作用于括号中的所有内容。这样匹配的内容就是 3 个“1 到 3 位数后面跟一个‘.’”。最后一部分的内容是匹配一个 1 到 3 位数。

整体的式子匹配的内容可能是这样“23.233.241.1”，稍有经验的人就可以看出这是一个 IP 地址。当然这个式子还有缺陷，如其可匹配“380.433.878.999”，而这样的形式是一个错误的 IP 地址，匹配也没有什么意义。完善这个正则表达式，是留给读者的思考题，如果能写出一个正确匹配 IP 地址的正则表达式，说明已经初步掌握了正则表达式的相关概念，应该可以看得懂常用的正则表达式了。



这里介绍的正则表达式语法是基本语法。不同的环境下正则表达式可能在细

节上对语法有不同的规则，使用前应该阅读相关环境的文档。

5.5 小结

本章介绍了编辑文本时最使用的三款工具 gedit、vim 与 Emacs 的使用方法，以及正则表达式的相关概念。为了兼顾 Windows 用户的操作习惯，本章对于一款相对来说较易上手的 gedit 的软件介绍内容比较丰富。而 vim 与 Emacs 的内容，只要参照本章的顺序进行练习操作，相信很快也会熟练应用的。

第 二 篇

Linux 系统管理篇

- 第 6 章　Shell 基本命令
- 第 7 章　文件管理
- 第 8 章　软件包管理
- 第 9 章　磁盘管理
- 第 10 章　进程管理

第 6 章　Shell 基本命令

Shell 是 Linux 的重要程序，提供了用户与核心的通信功能。可以说 Linux 就是在 Shell 基础上发展起来的。要掌握 Linux 的使用，则必须先要熟悉 Shell 的命令。Shell 提供的是文本界面的操作方式，与之相对的另一种界面是图形界面，提供了图形化的操作方式。从某种意义上讲，Shell 相当于 Windows 的命令行窗口，但功能要强大得多。本章主要介绍 Shell 的一些基本系统管理命令。

6.1　Shell 简介

Shell 是一个命令行解释器，是介于操作系统内核和用户之间的一层壳（Shell 的英文含义就有壳的意思，这也是为什么取名叫 Shell 的原因）。Shell 虽然不是系统内核的一部分，但其无疑是 Linux 操作系统最重要的组成部分。深入了解 Shell、熟练掌握相关命令，是用好 Linux 的关键。Shell 既然如此重要，对于其相关知识和发展历史及主要特性就不能一无所知了。

6.1.1　Shell 的种类

Shell 虽然看起来都一样，都提供了一个简单的文本窗口用来输入命令。但是，Shell 有许多种类。即使其基本功能一样，但不同的 Shell 功能各有特色。常用的有 Bourne Shell（简称 sh）、C-Shell（简称 csh）、Korn Shell（简称 ksh）和 Bourne Again Shell（简称 Bash）。

- Bourne Shell 是 UNIX 系统的默认 Shell，也是其他大多数 Shell 开发的基础。Bourne Shell 在编程方面表现优良，不过用户交互方面是其软肋。
- C-Shell 最大的优点是优秀的用户交互性能，其命令网络是 C 语言式的，这对于熟悉 C 的用户是个很大的福音，因为可以将 C 编程时的经验轻易移植到 Shell 中来。C-Shell 提供了很多现代化的功能，如命令补全、别名的使用等。
- Korn Shell 结合了 Bourne Shell 与 C-Shell 的优点，既具有强大的编程性能，又有良好的用户交互功能，而且其符合 POSIX 标准，所以可以在符合该标准的多种平台上正确运行。
- Bourne Again Shell（即 Bash）是 GNU 计划的一部分，是 Linux 系统默认的 Shell。
- POSIX Shell 是 Korn Shell 的一个分支。当前主要使用该 Shell 的是基于 HP 的系统。

Shell 对于系统来说，有着极其重要的意义。一般而言，Shell 是构建 Linux 类操作系统的基础。Linus 在开发 Linux 内核的最初版时，就努力地在内核上提供了 Shell，然后安装了 GCC。这样，其他众多的开发者就可使用 GCC 来扩展内核的性能以及开发其他可在 Linux 中运行的程序了。在众多开发者共同努力下，Linux 拥有了越来越多的可运行在 Shell 中的程序。最后与 X Winodow 兼容，拥有了图形界面，就是现在各大发行版图形界面的基础。提供 Shell 的操作系统所默认使用的 Shell 程序，如表 6.1 所示。

前面介绍的是系统底层的 Shell。对于用户来说，可能关心得最多的不是使用哪款底层的 Shell，而更关心的是调用 Shell 的终端程序是否好用，这些面向用户调用 Shell 的虚拟终端程序

如下所示。

表 6.1　各操作系统默认的 Shell

操 作 系 统	提供的默认 Shell
Linux	Bourne Again Shell
AIX	Korn Shell.
Solaris	Bourne Shell.
FreeBSD	Bourne Shell
HP-UX	POSIX Shell

- urxvt：一个小巧的终端。rxvt-unicode 还支持“服务器/客户端”的运行模式。
- urxvtd 启动一个守护进程 daemon（支持控制台）。
- urxvtc 启动客户端 client。多个客户端可以同时连接到一个 urxvtd，以达到节省系统资源的目的。
- mlterm：小巧的虚拟终端，内存占用少，速度快，支持多种语言。
- fish：自动补全功能较强、支持语法高亮，而且 fish 的键绑定也与标准 Bash 非常相似。
- gnome-terminal：Ubuntu 默认的虚拟终端程序，功能强大。但同时其体积也比较大，占用系统资源比较多，所以很多时候被其他虚拟终端所替代。

6.1.2　Shell 的作用

Shell 有两种工作方式，一是作为命令行提供给用户，二是作为编程语言供开发者使用。在作为命令行时，其功能是解释用户的输入，将其传送给系统，由系统来执行，并可以将执行结果或输出反馈给用户。作为编程语言时，Shell 可以像普通的高级语言一样进行程序设计工作。

1．命令行方式

Shell 识别用户操作的关键是用户输入的命令。一般来说，一条命令由一系列系统可识别的字符组合组成，字符中间可能由空格或 TAB 键隔开，并以回车符作为命令执行的信号。如 sudo gvim /usr/share/php5/php.ini。Shell 为用户提供了一个向 Linux 发送请求以便运行程序的系统级接口，Shell 可以调用系统内核来完成各种任务。

Shell 的主要功能之一是在交互方式下解释从命令行输入的命令。Shell 解析命令行，将其分解为词（也称为 token），词之间由空白分隔，空白可以是制表符、空格键或换行符。如果词中有特殊的元字符，Shell 会对其进行替换。对命令行的处理结束后，Shell 开始运行它。

2．定制用户环境

Shell 的另一项重要功能是定制用户环境，这通常在 Shell 的初始化文件中完成。初始化文件中有很多定义，包括设置终端键和窗口属性，设置用来定义搜索路径、权限、提示符和终端类型的变量，设置特定应用程序所需的变量，如窗口、字处理程序和编程语言的库等。

3．编程语言

Shell 的另一个重要功能是用做编程语言。Linux 中有一种特殊的可执行程序，其内容是由一系列命令组成的纯文本文件，这些文件实际上就是使用“Shell 语言”编写的程序。这些文件一般来说会完成一些复杂的系统设置或完成一系列重复的工作。Shell 程序可用任意文本编辑器生成，不过 Linux 环境中也有一些辅助编程的工具，如 Emacs 的 Shell 编程模式等。由于 Shell

编程的各种优点，故其是 Linux 中高级用户最经常使用的功能。

4．Shell 的模式

Shell 提供了用户与操作系统之间通信的不同模式，分别是交互式模式与非交互式模式。

- 交互式模式：Shell 等待用户的输入，并且可以立即得到响应。被称做交互式是因为 Shell 与用户进行交互。该模式也是大多数用户非常熟悉的：登录、执行一些命令、退出登录等。
- 非交互式模式（Shell script）。Shell 不与用户进行交互，而是读取存放在脚本文件中的命令。脚本文件是精心编制的，用于执行特定功能的命令的集合。脚本可以使许多操作自动完成，以提高使用效率。

6.1.3 Shell 的启动

Shell 启动前，先读取/etc/bash.bashrc 文件和$HOME/.bashrc 文件，对 Shell 进行初步的设置。前者为所有的用户定制环境，后者为拥有$HOME 目录的用户定制环境，如果前后两个目录对环境有不同的设置，以用户目录里的.bashrc 文件为主。这些文件简单介绍如下。

- .bash_profile：该文件只被登录 Shell 读取，操作系统内启动的 Shell 只读取.bashrc 文件。
- .bashrc：该文件被启动的每个 Shell 读取。
- .bash_logout：bash 退出时执行该文件。

用户使用的是哪一种 Shell，可在用户管理的相关目录中设定。详细内容可以查看第 22 章用户与组管理的相关内容。



图 6.1　虚拟终端

在图形环境下启动的 Shell 是虚拟终端，启动方法是依次选择 Application|Accessories| Terminal 命令，则启动一个窗口，如图 6.1 所示。

如图 6.1 所示是在主目录中执行了“ls -l”后的输出。可以看到，输出的文件是有颜色的，这也是 Linux 的 Shell 的一个特色。以各种颜色来区分不同的文件种类，各个颜色对应不同的文件种类，其详细意义参见表 6.2。

表 6.2　Linux 中不同颜色的文件的意义

ls 命令显示出的文件的颜色	文 件 种 类
白色	一般性文件
绿色	可执行文件，如可执行的脚本文件，bin 文件等
浅蓝	符号链接文件
蓝色	目录
红色	压缩文件，如 tar.gz、zip、rar 文件等

正如前面所说，这不过是个虚拟终端。要启动真正的终端，最简单的方法是按下“Ctrl+Alt+F2”组合键。最后这个 F2，可以是从 F1 到 F6 中的任意一个（键盘最上面的一排 Fn 键），代表了第 1 号终端到第 6 号终端。比如，按下 Ctrl+Alt+F5 组合键，启动的是第 5 号终端。

系统启动时，默认会启动 7 个终端。第 7 号终端就是当前正在运行的图形界面的终端。前 6 个终端分别可用“Ctrl+Alt+Fn”来进入。登录的具体步骤如下。

（1）进入终端后，系统会出现 login 字样，提示输入用户名进行登录。输入用户名，然后

按下 Enter 键。

（2）系统会提示输入密码，输入密码，然后按下 Enter 键进行登录。



为了安全起见，Linux 系统的 Shell 在输入密码时，默认是没有任何回显信息的。也就是说，Shell 在输入密码时并不像其他系统一样会显示一些星号或其他符号来提示用户。这样别人即使偷窥，也不会知道你的密码长度，会增强密码的安全性。即使是没有回显，输入的密码还是会被系统正常接受的。

（3）用户名与密码都输入正确，就可以登录进入一个真正的终端了。在终端中，可以使用命令行程序进行系统管理了。本章所介绍的 Shell 操作，在虚拟终端和终端上均可正确执行。



进入终端后，要返回到图形界面的话，按下 Alt+F7 组合键就行了。至于为什么按 F7 键，是因为图形界面运行在第 7 号终端上。

6.2　Shell 基本技巧

在 Shell 中可以输入命令，让系统执行来完成相应的工作。但在开始输入命令前，还是应该先了解 Shell 有什么技巧可以简化操作、提高工作效率吧。有句古话说得好，“工欲善其事，必先利其器”，掌握了这些技巧，就像握着柄宝刀一样，Shell 操作会变得简洁高效许多。而且，更为重要的是，这些技巧是不怎么需要死记硬背的，用上一两次，觉得不错，下次自然会用，然后就越来越熟练。

6.2.1　自动补全

Shell 操作的技巧中，首先要提到的一个功能就是自动补全功能。之所以要首先提及这项功能，是基于以下两个显而易见的原因。

- 人的大脑并不是计算机，不可能记得每条命令或每个目录的准确字符。这时，自动补全就可以补充命令或目录的字符，保证准确无误。
- 如果一个命令或目录过长，经常输入无疑是对手指的毒害。这时，自动补全就可以节省输入字符数目，保护手指。

一般来说，自动补全可以分为自动补全程序名和自动补全目录两种方式。在实际操作中，这两种方式均是极为有用的，应该在使用时多加练习。下面分别进行简单的介绍。

1．自动补全程序名

自动补全程序名的前提是系统中安装了要补全的程序，而且此程序必须放置在 Shell 搜索的目录里。Shell 默认情况下搜索的目录包括“/bin”、“/usr/bin”、“/usr/sbin”、“～/bin”。只有放在这些目录里程序 Shell 才会自动补全其名字。Shell 补全目录的前提是系统中存在以用户输入的前几个字符为开头的目录。使用也很简单，在想使用自动补全的时候，按 TAB 键就行了。这样听起来有点不知所云，还是以几个例子来说明。

首先是一个补全程序名的例子，用查看网络信息的 ifconfig 命令来说明该功能对于程序名的自动补全。参照如下步骤进行。

（1）打开虚拟终端，在其中输入 ifc，这是该程序的前几个字符，终端应该为

```
westnorth@15:53 ～$ ifc
```



“westnorth@15:53～$ ifc”中，符号“$"与其之前的“westnorth@15:53～”组成了 Shell 的命令提示符，而“$”之后的内容为用户输入的 Shell 命令。

（2）按下 Tab 键。这时，系统会将其余的字符补充完整，在终端中出现的是完整的命令 ifconfig，像这样：

```
westnorth@15:53 ～$ ifconfig
```

（3）按回车键，该命令得到执行，获得网卡和宽带的相关信息。

```
westnorth@16:12 ～$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:8d:5c:13:64
          inet addr:192.168.1.1  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::250:8dff:fe5c:1364/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:18451 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16179 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16465283 (15.7 MB)  TX bytes:1955681 (1.8 MB)
          Interrupt:18 Base address:0xe000
```

但这个自动补全的命令会遇到其他情况。有时候输入命令按下 Tab 键，并没有自动补齐。下面的步骤是以日历程序 calendar 来说明这种情况的。

（1）在终端中输入 cal，然后按下 Tab 键。Shell 并没有补全这个命令，是 Shell 的自动补全功能出现了问题还是系统中本来就不存在这个程序？当然，Ubuntu 不可能没安装此程序。那么问题的根源是什么呢？应该怎么来处理这样的情况？

出现这种情况的原因是按下 TAB 后，Shell 去寻找自动补全的可能目标，但却发现了不止一个目标，Shell 不知道用户到底想输入的是哪一个命令，于是就不可能自动补全了。

（2）想知道 Shell 发现了哪些以 cal 开头的程序，按两下 Tab 键，则 Shell 出现的内容类似于这样：

```
westnorth@16:14 ～$ cal #在命令后按两下 TAB 键
cal              calendar       caller
cal3d_converter  calibrate_ppa
westnorth@16:14 ～$ cal #显示可能的程序列表后自动将输入的内容继续显示在 Shell
```

（3）看来，给 Shell 的信息还不够多，以 cal 开头的程序有 5 个呢。没关系，再输入一个字母 e，像这样：

```
westnorth@16:14 ～$ cale
```

（4）现在按 TAB，以 cale 开头的程序只有 calendar 一个，Shell 不再迷惑，直接将其送上，就可按下 Enter 键执行了。



如果想知道系统可运行的所有程序名的列表，可以在 Shell 提示符后不输入任何字符，直接按两次 TAB 键。不过一般这样做是没有什么意义的，应该先输入

几个前导词限定命令的范围。

2．目录自动补全

功能与程序名自动补全相似。但其在实际应用中可能使用频率更高，因为目录自动补全不止在 Shell 中可以使用，在许多软件中也是可以使用目录自动补全来简化输入的。

要实现目录自动补全，目录应该跟在一个命令后面，而且该命令可以对目录或目录中的文件进行操作。以一个显示文本文件的 cat 命令来演示一下如何进行目录自动补全。此次要显示的文本文件是源文件列表，位置在“/etc/apt/sources.list”。具体步骤如下所示。

（1）在终端命令提示符后输入“cat /e”，像这样：

```
westnorth@16:41 ~$ cat  /e
```

（2）现在，按下 TAB 键。注意到了吗，目录自动补全功能开始工作，将目录补全为/etc/。现在终端应该类似这样：

```
westnorth@16:41 ~$ cat  /etc/
```

（3）继续输入下级目录开头的字母“ap”。输入这两个字符后，按 TAB，没反应。看来/etc 下有许多个以 ap 开头的目录，按两次 TAB 键，果然显示出了许多以 ap 开头的目录。输出如下所示：

```
westnorth@16:41 ~$ cat /etc/ap
apache2/    apm/        apparmor/   apparmor.d/ apport/     apt/
westnorth@16:41 ~$ cat /etc/ap
```

（4）再输入“t /sour”，命令行内容成为“cat /etc/apt/so”，再按 TAB 键，看能不能补全。运气不错，完整的文件名被自动补全功能填充在命令行后面，如下所示。

```
westnorth@16:48 ~$ cat /etc/apt/sources.list        #自动补全后，按下 Enter 键执行命令，输出如下所示
    # deb cdrom:[Ubuntu 8.04 _Hardy Heron_ - Release i386 (20080423)]/ hardy main restricted
    #deb cdrom:[Ubuntu 8.04 _Hardy Heron_ - Release i386 (20080423)]/ hardy main restricted
    # See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
    # newer versions of the distribution.(后面的内容略)
```

通过前面的示例，可以看到自动补全的强大功能。在实际的 Shell 操作中，也要记得经常使用自动补全功能，一方面可以提高效率，另一方面可以保证输入正确。

6.2.2　使用别名

别名即 alias，为便于理解，可以将其看成是一个命令的快捷方式。Shell 的使用过程中，许多命令后都要跟一些参数。对于常用的参数如果每次都是手动添加就显得比较麻烦，就可以通过别名功能来减少输入，提高效率。别名的使用比较简单，使用方法是：

```
alias 目的命令 = 源命令
```

- 目的命令：自己喜欢的而且容易记忆的字符串，如 home 等。
- 源命令：实际执行的程序。如果是有参数的命令，则必须用单引号括起来。这个命令比较复杂可以是带参数的命令，也可以是一些命令的集合，命令与命令之间用分号隔开即可。

定义了别名后，在 Shell 中使用别名，就相当于使用后面的源命令。使用时有两种方法，交互式与非交互式。下面分别予以介绍。

1．交互式

在 Shell 中直接输入 alias 命令。这种方式比较简单直观，但是只能在当前终端有效，比较适合于对当前终端中使用频率高的命令进行简化。但缺点是当终端关闭后，这些别名就失效了。下面以一个名为 book 的别名演示这些操作。具体步骤如下。

（1）打开终端，输入以下内容（功能是指定了一个名为 welcome 的别名，其实际执行的操作是更改目录到 Examples，然后打开 case_Wellcom.pdf 文件）。

```
alias welcome='cd Examples; evince case_Wellcome.pdf'
```

（2）按下 Enter 键执行该命令。

（3）在终端中输入 welcome，会打开一个 Ubuntu 的欢迎文档。

（4）关闭此虚拟终端，再重新打开一次虚拟终端，其实也就是重启了终端。

（5）在新终端中输入 welcome，按下 Enter 键运行，会提示“没有找到相关的程序名”。说明此别名已经不存在了。

2．非交互式

非交互式是指将 alias 的命令写入配置文件中。这个方式稍微复杂些，但别名是长期有效的，在终端关闭，甚至系统重启后该别名仍然可以使用。对于普通用户来说，这种方式是最常用的。

至于别名写入哪个配置文件，要看需求而定了。如果只希望特定用户可以使用这些别名，则将别名写入该用户主目录下.bashrc 文件。一般来说，如果 Linux 桌面用户就只有自己，那么简单地将别名写入当前主目录下.bashrc 文件（～/.bashrc）即可。如果希望此别名是全局性的，也就是说只要登录到本计算机的用户均可使用，可将别名写入/etc/bash.bashrc 文档中。Ubuntu 系统主目录的.bashrc 文件里就有一些默认别名，如下所示。

```
alias ls='ls --color=auto'
## ls --color=auto 是显示文件的颜色，以从颜色上区分文件的种类。这个功能被赋予到 ls 上，所以用 ls 显示的文件是有颜色的
#alias dir='ls --color=auto --format=vertical'
##ls --color=auto --format=vertical，是让 dir 命令也显示颜色。这条命令默认是被注释掉了的。注释一个别名是在其前面加一个#号，而启用别名则是将这个#号删除即可
#alias vdir='ls --color=auto --format=long'
##第三行将长格式带颜色列出文件命令赋予 vdir
alias grep='grep --color=auto'
##让 grep 默认显示颜色
```

别名是可以进行嵌套的，也就是说前面定义的别名，后面就可以当成命令来使用。如下的代码就是正确的。

```
alias g='gvm'              #定义了一个名为 g 的别名，其实际操作是打开 gvim
alias sg='sudo g'          #嵌套了上一个语句定义的别名，定义了一个名为 sg 的别名，是用 root 身份打开 gvim
```

别名虽然可以在很大程度上方便用户，但“过犹不及”，如果在实际使用中滥用别名，那么不仅会使得使用不便，反而会带来许多麻烦。以下是几点在实际使用中要注意的事项。

- 一般而言，别名是对经常使用而且输入字符太多的命令简化用的。
- 不要使用太多别名，造成记忆上的负担，而且没有什么实用价值。
- 对于软件有自己的配置目录的，尽量将对于该软件进行的相应别名操作写到软件的配置文件中。例如，有人用 alias server_name='ssh -v -l USERNAME IP ADDRESS' 对 ssh 的操作进行简化，其实，最好的方法是使用～/.ssh/config 文件做配置。
- 尽量不要对图形界面的操作使用别名。例如，有人使用 alias fftr='/home/jlwallen/firefox/firefox http://linuxtoy.org'标记书签。这就没有什么意义，还要在命令行输入 fftr，可能的后果是多打开一个 firefox，多占用一些系统资源，还不如直接加入到 firefox 的书签工具栏中实用些。
- 对于长目录的别名，最好是建立一个符号链接。
- 使用 alias 不加任何参数可以查看当前定义的所有别名。

笔者在实际工作中，得到了一些 alias 技巧，同时加上在其他资料中得到的技巧，将其一并列在表 6.2 中，可供用户按需要选择使用，以提高使用效率。

表 6.2　别名技巧

别　　名	实现功能
Alias cd..='cd ..'	这个看起来有点在奇怪，前后一样？其实后面多了个空格。在实际使用中，经常会发生忘记输入空格的情况，这个别名相当于容错机制，可以借鉴。但这种别名不推荐经常使用，因为有可能会养成输入错误的习惯
Alias l.='ls -a .* -d'	只显示隐藏文件，这是别名最能发挥作用的地方，命令长而且比较难以记忆，建议看到技巧性强而且比较长的命令就将其加入到 alias 中
Alias rm='rm -iv'	删除前提示，以更加安全地执行删除操作。这是出于安全方面的考虑，但会造成使用的不方便，只能在安全与方便中得其一而失其一了
Alias scilab=' ～ /software/download/scilab5.0rc2/bin/scilab.bin'	这也是比较常见的别名技巧，将一些目录比较深的软件用别名提升到顶层，直接输入名字即可运行
Alias ..=" cd .."	到上级目录，如果从 Windows 中转来，可能对 Linux 的方式不太熟悉，用别名来过渡一下不失为一个方法，但不宜长时间使用此方法
alias lla='ls -la'	显示文件详情，显示所有文件（包括隐藏文件）
alias is='sudo apt-get install'	安装软件
alias update='sudo apt-get update'	更新源
alias fm='free -mt'	显示内存使用情况
alias pt='pstree'	以树的方式显示当前进程
alias pa='ps aux'	显示当前运行进程的详情
alias pe='ps -e'	显示当前运行的所有进程
alias convmv-utf8=" convmv -f gbk -t utf-8"	将编码由 gbk 转换为 utf-8
alias df='df -h'	以人类可以识别的方式（MB 或 GB 为单位）查看磁盘的空间

6.2.3　查看最近使用过的命令

历史命令是系统保存的、在 Shell 中最近使用过的若干条命令。Ubuntu 会保存 500 条最近使用的记录。现代操作系统一般均提供了对历史命令的操作，原因是根据统计概率，人们经常

使用的命令集中在几十条上，而且使用过的命令在短期内使用的概率极高。所以，Linux 提供的历史命令理论上也可以提高使用效率。



学习过操作系统的朋友可能会对这个说法感觉熟悉。操作系统中高速缓存中的内容组织就是根据这个原则设定的，这样可以大大提高缓存的命中率，提高系统性能。

使用历史命令的最简单的方法就是在 Shell 中按上下方向键。按向上键，出现最后使用过的命令。再按向上键，出现倒数第 2 次使用过的内容，依此类推。看起来很简单，但实际上历史命令的使用有很多方面的内容。

最近使用过的命令会保存在 Shell 的记录中。要查看历史记录，在 Shell 输入“history”（英文意思是历史）命令，回车执行。输出可能类似于这样（本小节相关内容的介绍均以该输出为准）：

```
.....
  399  sudo apt-get install nfs-kernel-server
  400  sudo gvim /etc/default/portmap
  401  sudo gvim /etc/exports
  402  sudo gvim /etc/host.conf
  403  sudo gvim /etc/hosts.deny
  404  sudo gvim /etc/hosts.allow
  405  sudo /etc/init.d/portmap restart
  406  sudo gvim /etc/exports
  407  sudo exportfs -r
  408  sudo /etc/init.d/nfs-kernel-server restart
```

前面的 3 位数字是历史命令的编号，很快就可以看到这些编号也是历史命令里一个重要的部分，有其不可替代的作用。最后一行的 3 个数字是最后一次执行的命令的编号，本次输出结果表明最后一次运行的命令编号是 408，命令内容是 sudo /etc/init.d/nfs-kernel-server restart。跟在数字后面的字符是实际输入的命令，至于这些命令能不能实际运行，那就不是历史命令所关心的范围了。其关心的只是忠实地按照设置，记录历史命令，不评论、不篡改。从这些特性上看，history 这个命令倒也无愧于其名字了。

6.2.4 调用历史记录中的命令

在 Shell 中，按向上方向键，则最后一条命令（本例中是 408 号命令）出现在提示符后面，以供选用。要注意的是用方向键使用历史命令的功能不像补全功能那样按照前面输入的字符来匹配历史命令，而是简单地用前面的命令来替代当前行的内容。使用历史命令中的命令，也有许多方法，下面是一些例子。

1．按向上或向下方向键

把历史命令列表想象成一个台阶，按上方向键是向上走一阶（命令编号减 1），而按下方向键向下走一阶（命令编号加 1）。走到哪一个台阶，该台阶上的命令就会出现在命令提示符后面。向上或向下可以随意。如果双手不愿离开键盘的字母区时，可以按下 Ctrl+P（向上）组合键或者 Ctrl+N 组合键向下切换命令，功能与方向键一样。

2．用!!执行最后一条命令

如果不想使用方向键来引用命令记录。还可以使用“!!”来引用最后一个命令。

3．用!-1 也可以执行最后一条命令

要引用最后一条命令，还可以使用!-1，功能与上述两个方法是一致的。

4．用!gvim，执行最近执行的以 gvim 开头的命令

使用“!特定字符”不仅可以执行以 gvim 开头的命令，随着“特定字符”的不同，执行的命令也不同。如“!sudo apt-”就执行 399 条命令（参见 6.2.3 节 history 命令的输出理解）。

5．用!N 来执行编号为 N 的命令

如要执行 404 号命令，简单地输入!404，就可以执行 sudogvim/etc/hosts.allow 命令。

6．取得上一条命令的参数

用!$可以获得上一条命令的最后一项参数。如下面的用法：

```
westnorth@20:31 ～$ gvim .Xresources .vimrc .bashrc
westnorth@20:31 ～$ vi !$
vi .bashrc
```

第一条命令使用 gvim 来编辑.Xresources，.vimrc，.bashrc 文件，第 2 条命令 vi 提取上条命令的最后一项参数，第 3 行是回显区，表明执行了 vi .bashrc 操作，提取第 1 条命令的参数成功。这条命令只可以提取最后一项参数，这个技巧的等效方法是使用!!$，不过应该没多少人会舍易求难吧。

用!^ 可以获得上一条命令的第一条参数。

7．取得特定命令的指定参数

在 history 中，还可以取得某个以特定字符开头的命令的指定参数，如 gvim 开头的命令的第 2 个参数。示例如下：

```
westnorth@20:42 ～$ gvim .bashrc .vimrc #使用 gvim 编辑两个文件
2 files to edit
westnorth@20:43 ～$ cat !gvim:2      #取得以 gvim 开头的命令的第 2 个参数
cat .vimrc                           #显示了 gvim 编辑的第二个文件的内容
```

也可以使用!gvim:$来取得 gvim 命令的最后一项参数，使用!gvim:^来取得 gvim 命令的第一项参数。

6.2.5　history 的使用技巧

history 是一个比较复杂的命令，可以使用不同的参数来达到不同的功能。对于 history 来说，还可以设置一些系统变量来控制其行为。下面就来介绍一下 history 的一些设置及使用技巧。

1．立即更新命令列表

Bash 执行命令时不是马上把命令名称写入 history 文件的，而是存放在内部的 buffer 中，等 Bash 退出时会一并写入。调用 history -w 命令可要求 Bash 立即更新 history 文件。

2．使用 HISTTIMEFORMAT 显示命令执行时间

从命令行执行 history 命令后，通常只会显示已执行命令的序号和命令本身。如果你想要查看命令历史的时间戳，那么可以执行 export HISTTIMEFORMAT='%F %T '，然后再输入 history，输出可能类似于这样：

```
496  2008-10-05 21:15:09 sudo chmod -r a+rx *
497  2008-10-05 21:15:09 sudo chmod a+rx *
498  2008-10-05 21:15:09 sudo mkfontscale
499  2008-10-05 21:15:09 sudo mkfontdir
500  2008-10-05 21:15:09 sudo fc-cache -fv
501  2008-10-05 21:15:25 vi vi
502  2008-10-05 21:17:36 cd /home/westnorth/.fonts/
503  2008-10-05 21:17:40 gvim .bashrc .vimrc
504  2008-10-05 21:17:52 history|more
505  2008-10-05 21:18:10 history
```

本例设置环境变量在 501 条命令前，可以看出在此之前的 Bash 命令（小于 501）显示的时间是设置环境时的时间。之后的那些新执行的 Bash 命令才被打上正确的时间戳。

3．使用 Ctrl+R 组合键搜索命令历史

Shell 中 Ctrl+R 是极其有用的一个组合键。此组合键可以对命令历史进行增量搜索，输入的信息越多，搜索的结果越准确。

使用方法也很简单，在 Shell 中直接按 Ctrl+R 组合键，出现动态搜索提示。输入要搜索的内容，可以是命令名或命令名中的一部分，也可以使用的参数。发现输入不了任何内容的时候，说明历史记录中没有与输入的字符相匹配的结果。如果输入的字符找到了一条命令，但不是想要的那一个，可以再按 Ctrl+R 组合键搜索下一个匹配的命令。找到命令后，直接回车执行，或按向右方向键进行修改。

4．使用 HISTSIZE 控制历史命令记录的总行数

将下面两行内容追加到 bash_profile 文件并重新登录 bashShell，命令历史的记录数将变成 1000 条。

```
vi ～/.bash_profile
HISTSIZE=1000 HISTFILESIZE=1000
```

5．使用 HISTFILE 更改历史文件名称

默认情况下，命令历史存储在～/.bash_history 文件中。将下列内容添加到.bash_profile 文件并重新登录 bashShell，将使用.myhistory 来存储命令历史。

```
vi ～/.bash_profile
HISTFILE=/home/westnorth/.myhistory
```

6．使用 HISTCONTROL 从命令历史中剔除连续重复的条目

如果命令被连续执行了三次。执行 history 后你会看到三条重复的条目。要剔除这些重复的条目，可以将 HISTCONTROL 设置为 ignoredups：

```
export HISTCONTROL=ignoredups
```

7．使用 HISTCONTROL 清除整个命令历史中的重复条目

上例中的 ignoredups 只能剔除连续的重复条目。要清除整个命令历史中的重复条目，可以将 HISTCONTROL 设置成 erasedups，如下所示。

```
export HISTCONTROL=erasedups
```

8．使用 HISTCONTROL 强制 history 不记住特定的命令

在不想被记住的命令前面输入一个空格，就强制 history 忽略该命令，如下所示。

```
export HISTCONTROL=ignorespace
```

9．清除所有的命令历史

如果要清除所有的命令历史，可以执行如下命令。

```
history -c
```

10．使用 HISTSIZE 禁用 history

要禁用 history，可以将 HISTSIZE 变量设置为 0，或修改/etc/profile 将 HISTSIZE=1000 改成 0 或 1，或者清除用户 home 路径下.bash_history。

将 HISTSIZE 设置为 0 的方法如下所示。

```
export HISTSIZE=0
```

11．使用 HISTIGNORE 忽略历史中的特定命令

export HISTIGNORE=”cd:ps:top:”，就将 cd、ps、top 的命令执行不记录在案。当然，可以将这些命令更换为任何你不想在历史命令中记录的程序名。

12．命令中的字符串替换

history 在使用中，还有一种比较有用的技巧，就是可以替换最后一个命令中的特定字符。如“^me ^you”就将最后一条命令中的字符“me”替换为“you”。其一般格式如下：

```
^string1^string2          #恢复最后的命令，并用 string2 替换 string1
```

6.2.6　键绑定

Bash 底层采用 GNU 的 Readline 库作为底层输入的接口。Readline 支持两种行输入模式 Emacs 风格和 vi 风格，默认情况下使用 Emacs 风格。

1．键绑定表

表 6.3 是默认情况下的键绑定情况。同时，Linux 下最著名的 Emacs 编辑器，也是这种风格的键绑定。所以，如果对于 Emacs 比较熟悉，可能这些组合键已经烂熟于胸了。

表 6.3　Bash 组合键列表

绑定的键		实现的功能	说　明
Ctrl 区	Ctrl + A	到行的开始	
	Ctrl + B	向前移动一个字符	
	Ctrl + C	中止一个命令	
	Ctrl + E	删除当前光标处的字符	
	Ctrl + D	到行的结束	
	Ctrl + F	向后移动一个字符	
	Ctrl + K	从当前光标删除到行的结束	
	Ctrl + L	清屏	
	Ctrl + R	搜索历史命令	可以输入命令的几个字符，然后按 Ctrl+R 组合键进行搜索，可以大

			大缩小搜索范围
	Ctrl + U	从当前光标删除到行的开始	

续表

绑定的键		实现的功能	说　明
Ctrl 区	Ctrl + XX	在行的开始与当前光标切换光标位置	
	Ctrl + S	冻结 Shell	
	Ctrl + Q	解冻 Shell	在解冻 Shell 后，冻结期间输入 Shell 的命令会全部显示出来
	Ctrl + Z	挂起或停止正在运行的程序	
Alt 区	Alt + <	取历史命令中的第一条命令	不是所有的虚拟终端程序都支持 Alt 区的这些组合键，如 mlterm 就不支持这些组合键，但系统自带的 gnome-terminal 虚拟终端是支持这些操作的
	Alt + >	取历史命令中的最后一条命令	
	Alt + ?	显示当前目录中以输入字符开头的文件名列表	
	Alt + *	插入所有可能自动补全的结果	
	Alt + /	自动补全文件名	
	Alt + .	取得上个命令的参数	
	Alt + D	删除一个字符	
	Alt + L	将其后的所有字符变为小写	
	Alt + N	用非递增法搜索历史命令	
	Alt + P	反向用非递增法搜索历史命令	
	Alt + U	将其后的所有字符变为小写	
	Alt + Backspace	从光标处删除到行开始处	
2T 区（2T 的意思是按 TAB 键两次）	2T	所有可能的命令	
	（string）2T	所有以（string）开始的命令	
	/2T	包括隐藏目录的所有目录	
	*2T	当前目录中不包括隐藏目录的所有目录	
	～2T	"/etc/passwd"里的所有用户	
	$2T	所有的系统变量 All Sys variables	
	@2T	"/etc/hosts"里所有的变量	
	=2T	输出当前目录的文件和目录	



这些键绑定强烈建议您记住一些最常用的，以便大幅度地提高操作效率。由于这同时也是 Readline 控件的键绑定，故任何使用 Readline 控件的程序中都可以使用。例如 Bash、lftp、GDB 等程序。

2．Shell 的控制变量

Shell 提供了一些选项来控制其特点，可以输入“set -o”来显示当前 Shell 中可用的选项，得到的输出可能像这样：

```
westnorth@23:05 ～$ set -o        #显示当前 Shell 中可用的选项
allexport       off
braceexpand     on
emacs           on
errexit         off
```

```
errtrace        off
functrace       off
hashall         on
histexpand      on
history         on
ignoreeof       off
interactive-comments    on
keyword         off
```

前面的表示选项内容，从单词中可以得到其功能的大概意思。如输出的第 3 行中 emacs 表示键绑定采用 Emacs 模式，后面的标志 on 表示该模式是打开的。如果为 off，则表示该模式是关闭的。根据 Bash Shell 版本的不同，其中的一些选项也稍有不同（可能具有不同的默认值）。

通过输入命令 set -o option-name，可以打开该选项。而输入 set +o option-name，就可以关闭该选项。例如，可以使用 set -o emacs 来打开 Emacs 键绑定选项。可以用 bind -p 来看看系统中已经做了哪些键绑定，其输出的一行内容格式应如下所示。

```
"\C-r": reverse-search-history
```

这个绑定的意思是，在命令行进行历史记录查询的时候，可以用组合键 Ctrl+R 来进行反向搜索历史记录。该个快捷键前面已经使用过。

6.2.7　Shell 的变量

Shell 变量有系统变量、环境变量等，最经常使用的主要是系统变量。本节列出了常用的系统变量与环境变量，可以在使用时查阅。

1．系统变量

下面是一些常见的系统变量。将其列于下，以供用户查阅。

- $n：命令的参数，n 表示参数的顺序。如用./vbox start ，则$1 内容为 start。
- $#：参数的个数，如./vbox start，则$#的值为 1。
- $0：程序名。如./vbox start，$0 返回 vbox。
- $?：上一个命令或函数的返回值。
- $*：将参数保存为一个字符串。以“参数 1 参数 2 ... ”形式保存所有参数
- $@：将参数保存为一个字符数组。以“参数 1”、“参数 2 ...”形式保存所有参数
- $$：当前程序的 ID 号。
- $!：　上一个命令的 ID 号。

2．环境变量

以下是 Shell 的环境变量，可供用户在使用时查阅。

- PATH：命令搜索路径。为了安全起见，当前目录并没有在 PATH 目录里，所以运行当前目录中的程序要显示指定。如果当前目录中有一个名为 vbox 的程序，应该像这样“./vbox”使用，直接使用“vbox”则系统会报错。
- HOME：用户的主目录。
- HISTFILE：历史命令文件。
- HISTSIZE：历史命令文件中最多包含的命令数。

- LOGNAME：当前用户名。
- SHELL：当前 SHELL 所在的目录。
- TERM：终端类型。
- LD_LIBRARY_PATH：库文件搜索路径。



环境变量均可用“echo $命令”查看。如“echo $SHELL”显示当前 Shell 所在目录信息。

6.3 重定向与管道

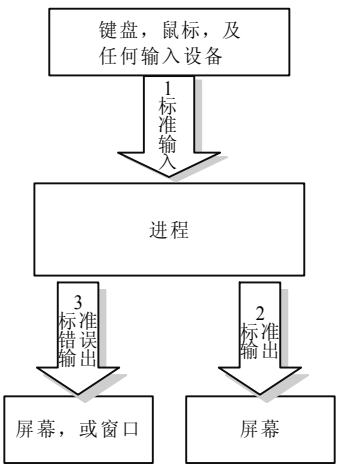


图 6.2 文件描述符

在 Shell 的使用中，管道与重定向功能是提高效率的重点所在。在实际使用这两个重要的功能前，先简单介绍一下其概念与原理。而为了了解重定向与管道，文件描述符也是必须要掌握的知识。

6.3.1 文件描述符

在 Linux 中，文件描述符是文件的基本属性。关于其概念，可以参看图 6.2。Linux 进程（除了守护进程）至少拥有三个文件描述符：标准输入（stdin）、标准输出（stdout）和标准错误输出（stderr）。

stdin 接收键盘、鼠标或任何输入设备的输入信息，文件描述符是 1；stdout 输出所有的正常输出结果，文件描述符是 2；stderr 输出的是所有的错误信息，文件描述符是 3。一般来说，这三个描述符与该进程启动的终端相关联。重定向和管道的目的是重定向这些描述符，如将 stdout 不输出到默认的屏幕，而是输出到一个文件中，或将默认的 stdin 更改为从一个文件读入数据，而不是从键盘或鼠标中读取输入。下面是重定向与管道的一些描述。

- 重定向：改变命令的输出方向。系统输出有标准输出和错误输出两种，默认是将这两者输出到屏幕上以供查看。但重定向可以改变这些，可以分别使标准输出定向到文件（将输出内容写入文件中），或者将出错信息定向到文件，以便查看。重定向的符号有>、<、>>、>&、>>&。
- 管道：将前一个命令的标准输出用管道输送到下一条命令中作为标准输入项执行。管道操作的符号是“|”。

6.3.2 重定向符的使用

前面已经介绍了重定向符的概念。本节主要介绍这些重定向符的使用方法与技巧。

1.“>”的功能

>的使用方法是“命令 > 文件”，该命令的意义如图 6.3 所示。

进程（命令会启动一个进程）接受标准输入后，该重定向符使得标准输出改变方向，不输

出到屏幕，而是输出到一个文件。但是标准错误输出仍然保持不变，输出到默认的屏幕或相应的设备。注意如果文件已经存在，则内容会被清空，然后写上新的内容。

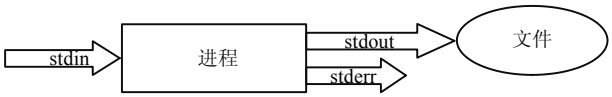


图 6.3　标准输出重定向

下面是一个使用该重定向符的示例，具体步骤如下：

（1）在 Shell 中输入 ls 然后运行，会出现当前文件列表。这是初学者都很熟悉的命令，很简单。

（2）使用该重定向符，使得 ls 的输出重定向到 file.list 文件中。在 Shell 中输入如下命令：

```
ls > file.list
```

这条命令的意思是将 ls 的输出不显示在屏幕上，而是重定向到 file.list（这个名称可以由自己设定）文件中。执行后，Shell 的屏幕上没有任何显示，说明重定向已经发挥作用。

（3）用 ls 命令查看一下，会发现当前目录中多了一个叫做“file.list”的文件，这个文件就是重定向的目标文件，命令的标准输出信息就写在里面。

（4）查看该文件的内容，使用如下命令：

```
vim file.list
```

会发现里面保存着“>”前面的命令“ls”的输出结果，也就是当前目录中的所有文件列表。

前面所述就是重定向符的功能。这个功能看起来简单得不值一提，但其实是很有用的。比如这个用法：

```
ls -l /dev/disk/by-uuid/>mydiskUUID
```

是将当前系统磁盘各分区的 UUID 信息写入一个叫 mydiskUUID 的文件中。当然，在其他许多方面这个功能还可以发挥更为重要的作用，这就依赖于使用者的想象力了。

2.“>>”的功能

>>可以看成是>的安全版本。因为>有一个问题，就是如果目标文件已经存在，>会将其清空，然后写入本次重定向的内容。这在有时候并不安全，很可能会错误地修改了重要的文件内容。这种情况就可以使用>>，其与>不同的是，如果文件存在，则将本次重定向的内容添加到文件的末尾。如果文件不存在，则建立一个文件。

这种方式可以在记录一些 log 信息时使用。如可以将 log 信息不断地添加到某一个文件中，以便将来查看。但缺点是这个日志文件会越来越大，如果 log 的数量过于巨大，就不得不用 cron 来自动备份了。本命令的其他功能与>是一致的，就不再重复说明了。

3.“>&”的功能

在使用前两个重定向符时要注意前两者均是标准输出的重定向符。如果想要将出错信息写入一个文件中，用前两个命令会发现错误输出信息仍然出现在屏幕上。要捕获标准错误输出，就要用到“>&”命令。其用法和前两条相类似：“命令>& 文件”。这个命令的功能如图 6.4 所示。

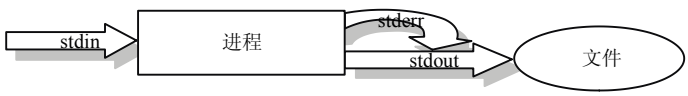


图 6.4　标准错误输出重定向

如图所示，“>&”的功能是将标准错误输出（stderr）重定向到标准输出（stdout），然后输出到一个文件中。可以用 asdf 这个系统中不存在的命令来学习一下这个用法。步骤如下所示。

（1）在 Shell 中输入 asdf（该命令不存在），得到错误提示，说没有此命令。

（2）用本重定向符将错误输出重定向到 mytest.error，使用如下命令。

```
adsf >& mytest.error
```

可以发现错误提示不见了。但是用 vi mytest.error 查看时，会发现原来得到的那个错误提示保存在这个文件中了。

这个功能至少可以使用在以下两种场所。

- 出错信息太重要，不容易记忆时可以到文件中查看出错信息，以便找到解决错误的方法。
- 出错信息无意义，没必要查看出错信息时，可以将其定向到/tmp/null 中让其消失。

4．“>>&”的用法

和第三个符号的用法一样。区别是如果后面跟的文件存在，就将重定向内容加到末尾，不存在，就创建该文件。

5．“<”的用法

如果一个进程需要得到输入信息的话，可以用这个重定向符让进程从文件中读取。这对于某些输入信息重复或输入信息有规律时很有用，用法是“命令 < 文件”。这样命令如果需要任何信息，就会去文件中读取。

6．“<<”的用法

“<<”也是输入重定向的一种。不过，其后跟的是定界符，这条命令使用方法有点特别，用一个例子来说明。在 Shell 中进行如下操作（以“#”开头的为注释）：

```
westnorth@22:53 ～$ <<myend
> oh,my God #注意看前面的提示符，不再是$，而是>，意思是 Shell 觉得输入的内容还不够
> so be it  #再输入一行，还是不够
> 这是什么    #再输入一行，还是不行
> myend     #最后，输入<<后面的字符串“myend”，回车后，输入才宣告结束
westnorth@22:53 ～$#输入结束，回到命令提示符
```

通过这个例子就可以知道“<<”的工作原理了。

7．完整的重定向命令

前面介绍的几个只是重定向符的简单使用，完整的重定向功能的用法如下：

```
command %n%c file。
```

参数的详细解释如下。

- command：要将其输出重定向的命令。
- %n：文件描述符，参见图 6.2，如果不指定该值，Shell 默认该值为 1，也就是标准输出。
- %c：重定向符，前面介绍的 5 个。
- file：重定向目标文件。

意思是将 command 这个命令的%n 文件描述符用重定向符%c 重定向到 file 中。这听起来

很难理解，看一下实际的使用方法就很容易理解了。

```
ls /noexist 2> ls_error
```

“ls /noexist”是命令，功能是显示一个不存在的目录，2 是文件描述符（2 是标准错误输出），>是重定向符，ls_error 是重定向的目标文件。整条命令的功能是将 ls 命令的出错信息重定向到 ls_error 文件中。Shell 对于书写格式也有讲究，注意在文件描述符和重定向符间不能有空格，否则本例中 Shell 不会将 2 解析为标准错误输出的文件描述符，而将其作为要写入 ls_error 中的一个字符。如果在 2 和>之间有一个空格，ls_error 的内容就只有这个数字 2，而不是标准错误输出的内容。

8．重定向的技巧

实际使用中重定向是很复杂的，可能会超出这个公式的范围。下面给出一些使用重定向时的技巧，可以在使用时参考。

- 将标准错误重定向为标准输入 command 2>&1。
- 将第一个文件既作为标准输入也作为标准输出 command <>file。
- 把文件描述符 m 作为标准输入 command <&m。
- 把标准输出重定向到文件描述符 m 中 command >&m。
- 关闭标准输出 command 1>&-。
- 关闭标准错误 command 2>&-。
- 关闭标准输入 command 0<&-。
- 关闭标准输入 command <&-。

6.3.3　管道

在 Shell 中，可以用管道操作符“|”来连接进程。管道的作用是将一个进程的标准输出通过管道变为另一个进程的标准输入，后面的命令可以直接处理前面命令的输出。功能如图 6.5 所示。

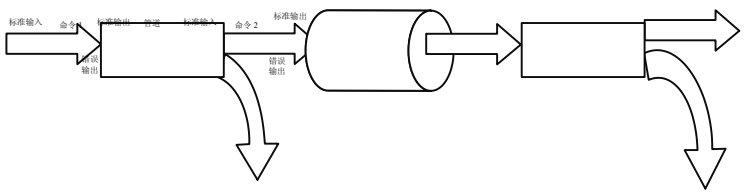


图 6.5　管道

如要查看系统安装的字体，可以用 xlsfonts 命令。但一般系统安装的字体比较多，所以 xlsfonts 的输出有许多屏幕的内容，很难看清其内容。如果要从其中过滤出想要的部分内容，就可以使用管道功能了。下面是两个使用实例。

例 1：显示系统安装的中文字体。

```
xlsfonts | grep gb2312
```

这样，显示出的内容就只有中文字体了。但可能经过过滤后，内容还是太多，屏幕要翻好几页，这里可以再使用前面介绍的重定向功能将结果输出到文件中。

```
xlsfonts | grep gb2312 >myfonts.cn
```

这样就可将系统安装的中文字体信息写入到 myfonts.cn 文件中。

例 2：显示特定的进程。

有时候需要要查看当前运行的特定名字的进程，可以用如下命令显示出 scim 输入法相关的进程。

```
ps -e | grep scim
```

了解管道的相关知识后，就应该介绍一下三通命令 Tee。这个命令很有意思，是将标准输出既显示在屏幕上，同时又写入一个文件。从逻辑意义上讲，是一个输入，两个输出，像管道中的三通管。其名字就是根据功能来取的，字母 T 就是一个三通的形状，而其发音是 Tee，因此而得名。使用很简单，继续沿用前面的例子，在命令行中输入如下命令。

```
xlsfonts | grep gb2312 | tee myfonts.cn
```

这样在显示字体的同时，将其写入了 myfonts.cn。使用管道连接的进程数目是没有限制的，也就是说，可以连续使用无数个管道符进行操作。

6.4 文件内容相关命令

在 Linux 中，基本上所有的设置文件均是纯文本格式，用任何文本编辑器均可编辑。这样做，感觉好像效率有点低，也不够酷，不太符合 Linux 是高手专用的这个假设。因为用十六进制编辑器编辑二进制文件才像是高手的做法。但这样做有许多好处，其中很重要也很明显的一条就是可以用屏幕编辑器 vi 或任意文本编辑器方便地更改各种程序的配置。否则就要使用专用的设置软件，或者不得不用反编译软件来查看设置的关键数据。所以，对于 Linux 的设置，纯文本文件内容的处理绝对是重要的基础。

6.4.1 输出文件内容

有时候，只是简单地要查看文本文件的内容。用文本编辑器打开有点麻烦，而且还有错误地将设置更改的风险。这时，显示文本文件内容的几个命令就可以发挥作用。

1．输出文本文件内容命令 cat

Linux 里有一只猫，这猫不是普通的猫，而是 cat（cat 的英文意思是猫）。功能是将文本文件的内容显示在屏幕上。最简单的使用方法是 cat file，如：

```
cat .bashrc
```

屏幕上会显示.bashrc 的内容。如果内容太多，一个屏幕显示不完，会自动滚动屏幕进行显示。这样的话，前面的内容就看不到了。所以可以使用前面介绍的管道功能将其重定向到另一个命令 more 中：

```
cat .bashrc | more
```

这样，显示一屏后，会自动暂停，等待用户单击空格后继续翻页。

cat 的完整用法如下：

```
cat [option] file
```

其中，option 是参数项，可以使用的参数如下：

- -b：给非空白行编上行号；
- -n：给所有行（包括空白行）编上行号；
- -s：多个连续的空行只显示一行；
- -T：显示制表符（TAB 键输入的内容）；
- -v：显示非打印字符；
- -E：显示行尾符；
- -A：相当于同时使用 vET。

例 1．cat 参数的使用示例。

使用“cat -An .bashrc”显示.bashrc 中制表符、显示非打印字符、显示行尾符、并且显示行号，输出类似这样（由于篇幅所限，只截取了部分内容）：

```
……
35  if [ -n "$force_color_prompt" ]; then$
36      if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then$
37  ^I# We have color support; assume it's compliant with Ecma-48$
   ^I# (ISO/IEC-6429). (Lack of such support is extremely rare, and such$
   ……
```

其中，前面的两位数字是行号（由-n 参数指定），^I 是制表符，行尾的$是行尾的标志，并不是文件本身的内容。

例 2．用 cat 创建文件示例。

cat 再加上重定向符，可以创建文件，如下所示。

```
westnorth@20:10 ~$ cat > cattest <<jieshu
> hello,world
> 你好，Linux 世界
> jieshu
```

将 cat 的输出重定向到文件 cattest，然后使用 jieshu 作为定界符，这样就可以在命令行中输入文本内容，然后输入 jieshu 表示输入结束。建立完成后，用 cat cattest 可以查看输入的内容（除了 jieshu 定界符以外的内容）。

例 3．用 cat 连接文件示例。

cat 的使用中，很重要的一个技巧就是用其将几个文件连在一起。如这样：

```
cat cattest1 cattest2 cattest3 > cattest
```

是将 cattest1、cattest2、cattest3 的内容依次加入到文件 cattest 中。注意这两例用法里的重定向符>，如果对前面介绍的还有印象的话，就明白这个重定向符会将原来的文件内容清空，写上新的内容。如果想保留原来的内容，应该使用>>重定向符。

2．进阶文本显示命令 more

猫还不够，要更多，用 more 吧。more 最简单的使用方法是“more .bashrc”，意思是以屏幕为单位显示文件的内容。实际上，其功能要比 cat 多，使用也要更为复杂。

其使用格式如下：

```
more [-dlfpcsu] [-num] [+/ pattern] [+ linenum] [file ...]
```

-dlfpcsu 是参数表，详情如下：

- -d：显示帮助信息；
- -c：从顶部清屏然后显示；
- -d：提示 Press space to continue，'q' to quit.（按空格键继续，按 q 键退出），禁用响铃功能。
- -l：忽略 Ctrl+l（换页）字符。
- -p：通过清除窗口而不是滚屏来对文件进行换页，和-c 参数有点相似。
- -s：把连续的多个空行显示为一行。
- -u：把文件内容中的下画线去掉。
- -num：指定每次显示的行数，如-8，则每屏显示 8 行。
- +/pattern：打开文件时，在其中寻找 pattern 字符串，其中 pattern 可以是任意字符。
- +linenum：从 linenum 行开始显示文档。

如果输入下列命令，其意思是从 34 行开始打开文件，并从 34 行开始寻找“set”，将 34 行后第一个包含“set”的行显示在屏幕上：

```
more  +/set  +34  .bashrc
```

使用 more 后，屏幕显示一些内容，然后在屏幕最下方显示一行“—More—（71%）”或类似内容。这样就进入了 more 的命令行模式中。可以按空格键下翻一页，或按回车键下翻一行来浏览文件内容。这两种方式是命令行模式中的命令，其他常用的 more 命令行模式操作键及功能如表 6.4 所示。

表 6.4　more 命令行模式操作键及功能列表

命令行模式操作键	功　　能
h 或 ？	显示帮助信息，包括命令的使用方法等
Q 或 q	退出
V	使用 vi 编辑文件
/pattern	查找 pattern，这个 pattern 可以是正则表达式
/	继续查找上次查找的内容
！<cmd>或:!<cmd>	执行外部程序
:p	向前滚动屏幕
:f	显示当前文件名和当前行
.	重复上一个命令
'	回到搜索前位置，搜索改变了当前位置后比较有用
Z、 Ctrl+D 或 d	功能与回车类似

这些键的功能与著名文本编辑器 vi 的使用极其相似，所谓“一法通，百法通”，熟练掌握了这些键的功能，对于使用许多软件都是大有裨益的。more 的长处是灵活、强大，但笔者觉得其不如 cat 的地方是不能整体显示行号，只能用=或“：f”显示当前一行的行号。

3．高阶文本显示命令 less

是 more 的进化版，与其名字不同的是，less 提供更多的选项。与 more 最大的不同是可以方便地让文本向前翻页或向后翻页（使用上下方向键）。less 在显示时只是先读出要显示的部分内容，因此，在对大文本内容读取时，其速度比文本编辑器速度要快些。其用法如下：

```
less [参数]  文件
```

常用参数如下。

- -c：从顶部（从上到下）刷新屏幕，并显示文件内容，而不是通过底部滚动完成刷新。
- -f：强制打开文件，二进制文件显示时，不提示警告。
- -i：搜索时忽略大小写，除非搜索串中包含大写字母。
- -I：搜索时忽略大小写，除非搜索串中包含小写字母。
- -m：显示读取文件的百分比。
- -M：显示读取文件的百分比、行号及总行数。
- -N：在每行前输出行号。
- -p pattern：搜索 pattern，比如在/etc/profile 搜索单词 MAIL，就用 less -p MAIL /etc/profile。
- -s：把连续多个空白行作为一个空白行显示。
- -Q：在终端下不响铃。

less 也有命令行模式，其命令融合了 vi 和 more 的内容，more 中可用的功能在 less 里也可以全部使用。具体用法可以参考前面介绍的 more 的相关内容。

4．显示文件前面若干行的命令 head

head 的功能是显示一个文件的内容的前多少行，用法比较简单。具体使用命令的格式如下：

```
head  -n  行数值  文件名
```

也可以将参数变为-c，显示前面的多少个字节。同时，可以将数字设置为负值，以不显示最后几行或最后几个字符。用法示例如下：

```
head -n 10 mytest   #显示 mytest 的前 10 行
head -c 10 mytest   #显示 mytest 的前 10 个字符
head -c -10 mytest  #显示除了最后 10 个字符外其余的内容。
```

5．显示最后若干行内容的命令 tail

tail 的功能是显示文件的最后几行文本，用法是比较简单的。

```
tail   -n  行数值  文件名
```

其基本用法与 head 类似，有几个参数有点意思，可以特别介绍一下。

- --retry：如果文件打开出现错误，可以一直重试，这个在网络中比较有用。
- -f ：可以跟踪文件的变动，后面还可以跟一些特别的指令。
- -pid [进程号]：跟在-f 后面，可以使 tail 在进程终止后自动退出。
- -s [数字]：跟在 f 后面，可以让 tail 休眠特定的时间，单位是秒，默认为 1 秒。

显示“.bashrc”文件从 23 行开始的内容，显示一次后休眠 10 秒再检查文件是否有变动，使用如下命令即可：

```
tail -n 23 -f -s 10 .bashrc
```

6．显示文本命令 pg

作用和其他显示文本内容命令相似，而且是原来大多数系统“man”命令显示内容时使用的命令。其在显示一屏后，会在屏幕下方出现一个命令提示行，输入命令可以翻页或进行其他操作。pg 命令有些子命令，将其列于下面。

- 回车：下一屏。
- l：向上一行。
- q：退出。
- /pattern：查找特定的字符串，可以使用正则表达式。
- /：继续上次的查找。

7．二进制方式显示文件内容命令 od

od（octal dump）命令可以以十进制、八进制、十六进制和 ASCII 码来显示文件或者流，在查看不能直接显示在终端上的字符或者显示存储在二进制文件中的数据信息时很有用。

使用语法：od ［选项］ 文件…

可用的选项的含义。

-A：指定地址基数，主要的参数如下。

- d：十进制。
- o：八进制（系统默认值）。
- x：十六进制。
- n：不输出位移值。

-t：指定数据的显示格式，主要的参数如下。

- c：ASCII 字符或反斜杠序列。
- d：有符号十进制数。
- f：浮点数。
- o：八进制（系统默认值为 02）。
- u：无符号十进制数。
- x：十六进制数。

除了选项 c 以外的其他选项后面都可以跟一个十进制数 n，指定每个显示值所包含的字节数。



od 命令系统默认的显示方式是八进制，这也是该命令的名称由来（Octal Dump）。但这不是最有用的显示方式，用 ASCII 码和十六进制组合的方式能提供更有价值的信息输出。

用法示例：

- od -t a file：以字符格式显示文件内容。
- od -c myfile：以 ASCII 格式显示文件的内容。
- od　myfile：以八进制格式显示文件的内容。
- od -A x myfile：以十六进制格式显示文件的内容 （内容与上面相同，只是偏移量不同）。
- od -A d myfile：以十进制格式显示文件内容。
- od -j 040 myfile：从第 40 个字符（以八进制）开始显示 file 文件的内容。
- od -N 10 myfile：只显示文件前面 10 个字符的内容（以八进制计算） 。

6.4.2　文本文件比较

在日常工作中，经常会用到文本比较的命令。如有时对程序进行了微小的改动，但别人需要查看对程序进行的哪些改动时，一行行查看代码很慢，就可以使用文本比较命令自动检查文件的改动在哪儿。

1．文本文件比较命令 diff

diff 的作用是比较两个文档，并且给出如何将源文档的一行内容轮换为目标文档的对应内容的方法，其使用方法是：

```
diff [options] 源文档 目标文档
```

option 是参数项，可取如下值。

- -a：将任何文档当作文本文档来处理。
- -b：忽略空格造成的不同。
- -B：忽略空行造成的不同。
- -c：使用纲要输出格式。
- -H：利用试探法加速对大文档的搜索。
- -I：忽略大小写的变化。
- -n --rcs：输出 RCS 格式。
- -p：显示更改发生于哪一个函数（仅限于 C 语言）。
- -r：递归处理目录中的所有子目录与文件。
- -e：为行编辑器 ed 生成脚本文件，ed 使用该文件就可将源文档转换为目标文档。

由 diff 产生的文档转化指令一般由“数字+一个字母+数字”组成，数字代表行号，一个字母代表一个文档的转化指令，整个序列代表了执行此操作后，源文档就与目标文档相同。其详细意义如表 6.5 所示。

表 6.5　diff 输出内容的意义

diff 产生的内容	如何将源文档转换为目的文档
23a34，45 >目标文档的第 34 到 45 行内容	将目标文档的第 34 到 45 行内容添加到源文档的 23 后
12，23c26，36 <源文件第 12 到 23 行内容 --- >目标文件第 26 到 36 行内容	将源文档的 12 到 23 行转换为 34、45 行内容
13，24d34 <源文件第 13 到 24 行内容	删除源文档的 13 到 24 行内容后就与目标文档第 13 行后内容一致

运行 diff 命令，如果两个文件不同，会产生一些输出。如果没有进行重定向，会将输出显示在屏幕中。如输出如下（序号是笔者所加）：

```
(a)23a34,36
>hello world
>你好
(b)12,13c13,14
<the fox jump the lazy cat
<dawning dog
---
>printf(“hello,world”);
>exit();
```

```
(c)13,14d13
<thank you
<my secret is …
```

（a）说明目标文档比源文档在 23 行后多了两行，后面列出的即是该两行的内容。

（b）说明源文档 12、13 行与目标文档 13、14 行存在着不同，后面分别将其列了出来。

（c）源文档将 13、14 行删除后就与目标文档 13 行后相同，后面列出的是 13、14 行内容。

输出中，前面有“>”的说明是引用自目标文档的内容，前面有“<“的说明引用自源文档的内容，而前面没有符号的，则是相关的文档处理命令。

diff 命令还有一个重要的用途是制作代码补丁。想象一下，对于一个开发者来说，下载一个开源的程序代码，然后对其做了比较小的改动后要再发布这些改动。这时，最好的做法不是发布所有的代码，而是用 diff 命令制作一个源代码的补丁，将其发布，供有需要的用户使用。

设源程序代码存放在目录“～/dev/pale”，而自己修改后的代码保存于目录“～/dev/mypale”中，就可以使用如下命令制作补丁。

```
diff  -rupN  ～/dev/pale/  ～/dev/mypale/  >  addlight.patch
```

这样，就制作了名为“addlight.patch”的补丁。其他用户下载该补丁后，就可使用 patch 命令给源代码打补丁，得到你所增添的新功能了。

2．二进制文件比较命令 cmp

cmp 也是比较文件内容的。如果两个文件内容相同，则 cmp 命令不显示任何内容。如果文件内容有差异，则 cmp 命令显示内容不同的第一个字节数和行数。一般用 cmp 命令比较二进制文件，而用 diff 命令比较文本文件。cmp 的用法如下：

```
cmp [OPTION]... FILE1 [FILE2 [SKIP1 [SKIP2]]]
```

OPTION 可取下列值。

- -b：输出文件发生不同的字节。
- -i num：越过开始的 num 字节开始比较，也就是说，开始的 num 字节不参与比较。num 是要越过的字节数。而且 num 可以使用字节长度单位，由小到大依次为 K、M、T、P、E、Z、Y，后者为前者的 1024 倍。
- -i num1:num2 越过 FILE1 开始的 num1 以及 FILE2 的 num2 字节开始比较。
- -l：显示每一个不同点的字节号（10 进制）和不同点的字节内容（8 进制）。
- -n num：设定比较的上限，最多比较 num 个字节。
- -s：如果文件相同，则只返回 0，文件不同，则只返回 1，发生错误则返回 2。

不带任何参数地运行 cmp，则比较两个文件。如果文件内容相同，则不显示消息；如果不同，则显示第一个不同的位置。

示例 1：不加参数地使用 cmp 命令比较两个文件用法如下所示。

```
$cmp file1 file2                          #比较两个文件 file1、file2
file1 file2 differ: char 23, line 6
#如果输出是这样，则说明 file1 与 file2 的内容在第 6 行的第 23 个字符开始有差异
cmp: EOF on file1
#如果输出是这样,则说明 file2 前半部分与 file1 相同，但在 file2 中还有其他数据
```

示例 2：如果加上-l 参数使用 cmp 命令，则比较文档，显示有差异的字节位置，然后显示该字节的内容。如下所示：

```
$cmp -l file1 file2          #使用 cmp 命令比较 file1 与 file2
34 75 53
#输出是这样，则说明在 34 个字节处两个文件开始不同，其中一个的内容是 75(八进制)，另一个是 53(八进制)
```

6.4.3　文本处理

文本处理包含的内容很多，如排序、替换、编辑等操作。本节主要介绍对文本进行编号的程序 nl、对文本进行排序的程序 sort，以及字符转换命令 tr。

1．文本行编号命令 nl

nl 是给文本文件每行加上行号，然后输出的命令，其用法如下所示。

```
nl [OPTION]... [FILE]...
```

OPTIONS 代表可用的选项，部分常用的选项如下。

- -b、--body-numbering=STYLE：自定义显示的行号的样式。
- -I、--page-increment=NUMBER：行号递增数，默认是 1，可以设置为其他数值。
- -n、--number-format=FORMAT 指定附加行数的格式（FORMAT）。
- -s、--number-separator=STRING：使用自定义的符号来区分行号与文本内容。
- -w、--number-width=NUMBER：指定行号所占的宽度，默认是 6 位。

而参数 b 可显示的样式，可以是以下几项。

- a：每一行都加上行号。
- t：只有非空白的行才加上行号。
- n：每一行都不加行号。

n 参数可指定的格式，可以是以下几项。

- ln：行号左对齐，前面不加 0 补齐。
- rn：行号右对齐，前面不加 0 补齐。
- rz：行号向右对齐，前面加 0 补齐。

下面是一些参数的用法示例（其中标明“本行无内容”的行是没有实际内容的空行，这是为了演示 nl 命令的功能特意留出的空行）。

示例 1：nl 默认行为，只给非空白的行加行号。

```
earthfans@mylaptop:~$ nl -bt .bashrc
#只有非空白的行才加行号，这是默认的行为
     1   # ~/.bashrc: executed by bash(1) for non-login Shells.
     2   # see /usr/share/doc/bash/examples/startup-files (in the package
bash-doc)
     3   # for examples
       本行无内容
     4   # if not running interactively, don't do anything
```

示例 2：设定行号的递增量为 3 的输出效果。

```
fans@mylaptop:~$ nl -i3 .bashrc        #设定行号递增量为 3
     1   # ~/.bashrc: executed by bash(1) for non-login Shells.
     4   # see /usr/share/doc/bash/examples/startup-files (in the package
bash-doc)
     7   # for examples
       本行无内容
    10   # if not running interactively, don't do anything
```

示例 3：给空白行添加行号的示例如下：

```
earthfans@mylaptop:~$ nl -ba .bashrc
#给空白行也添加行号
     1   # ~/.bashrc: executed by bash(1) for non-login Shells.
     2   # see /usr/share/doc/bash/examples/startup-files (in the package
bash-doc)
     3   # for examples
     4   本行无内容
     5   # if not running interactively, don't do anything
```

示例 4：使用特殊符号作为行号与实际内容之间的分隔符，示例如下：

```
fans@mylaptop:~$ nl -s +__+ .bashrc
#使用+__+作为行号与实际内容之间的分隔符
     1+__+# ~/.bashrc: executed by bash(1) for non-login Shells.
     2+__+# see /usr/share/doc/bash/examples/startup-files (in the package
bash-doc)
     3+__+# for examples
          本行无内容
     4+__+# if not running interactively, don't do anything
     5+__+[ -z "$PS1" ] && return
```

示例 5：设置行号的显示宽度，示例如下：

```
fans@mylaptop:~$ nl -w3 .bashrc          ##设置行号显示宽度为 3
  1 # ~/.bashrc: executed by bash(1) for non-login Shells.
  2 #  see  /usr/share/doc/bash/examples/startup-files  (in  the  package
bash-doc)
  3 # for examples
    本行无内容
  4 # if not running interactively,don't do anything
  5 [ -z "$PS1" ] && return
```

示例 6：设置行号前面用 0 补齐，示例如下：

```
earthfans@mylaptop:~$ nl -nrz .bashrc
#设置行号前面用 0 补齐
000001  # ~/.bashrc: executed by bash(1) for non-login Shells.
000002  # see /usr/share/doc/bash/examples/startup-files (in the package
bash-doc)
000003  # for examples
```

```
        本行无内容
000004  # if not running interactively, don't do anything
000005  [ -z "$PS1" ] && return
```

2．文本排序命令 sort

sort 命令的功能是对文件中的各行进行排序。如不指定排序的列（一行文本，用空格或 TAB 隔开的一串字符称为一列），则从行开始逐字符进行比较，然后进行排序。其完整语法如下：

```
sort [参数] 文件
```

参数可取下列值：

- -k：指定排序的列，后面跟一个数字，指定以某一列进行排序。
- -R：将各行进行随机排列。
- -t：指定列的分隔符（默认的分隔符是空格和制表符），这在使用了特定的分隔符时很有用，如列间用“|”隔开，就可用“sort -t '|' myfile”。
- -f：排序时忽略字母的大小写。
- -m：若给定文件已排好序，合并文件。
- -c：检查给定文件是否已排好序，如果没有排好序，输出出错信息，并以状态值 1 退出。
- -u：对排序后认为相同的行只留其中一行。
- -o：将排序输出写到输出文件中而不是标准输出，参数后跟输出的文件名。
- -d：按字典顺序排序，比较时仅字母、数字、空格和制表符有意义。
- -r：按逆序输出排序结果。
- -b：忽略前导的空白（空格和制表符）。

以下是一些例子。

- sort -k 3 myfile：以 myfile 的第 3 列为基准进行排序。
- sort -t '|' -k 3 myfile：将 myfile 的列分隔符定为“|”，然后以第 3 列为基准进行排序。
- sort -r -k 3 myfile：以 myfile 的第 3 列为基准进行排序，然后以反序输出。

3．转换字符命令 tr

tr 作用是进行单字符替换或删除等操作。如下所示命令将 myfile 文件中的 a 替换为 A：

```
tr 'a' 'A' < myfile
```

要注意是要转换的文件须用<进行重定向，而且替换是一对一的，如只能将 a 替换为 A 等。如果试图进行多字符的替换，可能会出现想象不到的错误。如果用户想用如下命令将 myname 替换为 westnorth：

```
tr 'myname' 'westnorth' <myfile
```

并不会将文件中的 myname 字符串替换为 westnorth。而是将所有的 m 替换为 w，将所有的 y 替换为 e，依此类推。

带有最常用选项的 tr 命令格式如下：

```
tr -c -d -s ["源字符列表"] ["目标字符列表"] <myfile
```

参数如下。

- -c：在“源字符列表”中未出现的字母用目标字符列表中的字母替代。

- -d：删除“目标字符列表”中所有输入字符，这里只用一个字符串参数就行了。
- -s：对于出现在“源字符列表”里的字母，如果发现其重复出现，就将其压缩到只有一个字母。
- input-file 是转换文件名，也可以使用其他格式输入。

用法示例：

- tr -s "[a-z]"<myfile >myfile_out：压缩 myfile 文件里面重复的小写字符，并将修改后的文本存入 myfile_out 文件中；
- tr -c“[A-S]”“*”<myfile：将 myfile 中不在 A 到 S 之内的所有字母均用“*”代替；
- tr -d“[d-m]”<myfile：将 myfile 中 d 到 m 的所有小写字母删除；
- tr“[:alpha:]”“*”<myfile：将所有的英文字母替换为“*”；
- tr“[:alnum:]”“*”<myfile：将所有的英文字母及数字全替换为“*”；
- tr 'A-M N-Z a-m n-z' 'N-Z A-M n-z a-m' filename：这是著名的 ROT13 加密法，将 26 个项字母首尾相连，然后将字符用其后第 13 个字符替换；
- tr 'N-Z A-M n-z a-m' 'A-M N-Z a-m n-z' filename：解密文件，使之可读，其实就是将前面两个字符串调换了一个位置。

使用时要注意，指定“源字符列表”或“目标字符列表”的内容时，对其格式也有限制，如下所示。

- [a-z]：a-z 内的字符组成的字符串。
- [A-Z]：A-Z 内的字符组成的字符串。
- [0-9]：数字串。
- \nnn：一个三位的八进制数，对应有效的 ASCII 字符。
- [O*n]：表示字符 O 重复出现指定次数 n。因此[O*2]匹配 OO 的字符串。
- [abde]：指定的一个字符序列，这个可在 ASCII 码范围内任意指定。

tr 中可以使用一些特定控制字符进行操作，这些控制字符也列在下面，“\”后面跟的 3 位数字与前面的有同样的效力，就是上面列表中的“\nnn”格式。

- \a、\007：铃声。
- \b、\010：退格符。
- \f、\014：走行换页。
- \n、\012：新行。
- \r、\015：回车。
- \t、\011：Tab 键。

而且，tr 还支持使用 POSIX 元字符，这样就可以简化操作。

- [:alnum:]文字数字字符。
- [:alpha:]文字字符。
- [:digit:]数字字符。
- [:graph:]非空字符（非空格、控制字符）。
- [:lower:]小写字符。
- [:cntrl:]控制字符。
- [:print:]非空字符（包括空格）。

- [:punct:]标点符号。
- [:space:]所有空白字符（新行，空格，制表符）。
- [:upper:]大写字符。
- [:xdigit:]十六进制数字（0～9，a～f，A～F）。

6.4.4　awk 文本处理程序

awk 是一种文本文件处理工具。awk 这 3 个字母看起来没有任何可以暗示其功能的意义，不过 awk 是三位程序大师首字符的缩写，他们分别是 Aho、Peter Weinberg 和 Brain Kernighan。正是这三位创造了 awk。

awk 可以通过扫描筛选符合条件的文本行，并生成报表。awk 提供了极其强大的功能：可以进行样式装入、流控制、数学运算符、进程控制语句甚至于内置的变量和函数。它还是一种文本处理语言，当处理的文件比较复杂时，可以使用 awk 语言编写成 awk 程序执行。awk 的三位创建者已将 awk 语言正式定义为“样式扫描和处理语言”，该语言复杂程度低于 C/C++等高级语言，但功能又强于 Shell 等脚本语言。所以，在很多场合有其用武之地。

1．awk 基本用法

awk 有 gawk 和 pgawk 两个版本，其基本使用方法是一致的，一般常用的是 gawk，最简单用法是在 Shell 中使用。用法如下：

```
awk ' pattern {action} ' file
```

参数说明：

- pattern：awk 可接受许多不同形态的 Pattern。一般常使用 "关系表达式"（Relational expression）来当 Pattern。比如“3>2”。
- action：awk 的指令，比如“print”。下文将给出详细的指令表。
- file：awk 处理的文本文件名。awk 不修改输入文件，如果未指定输入文件，它将接受标准输入，并将结果显示在标准输出上，其还支持输入输出重定向（<、>、>>、<<以及|）。

awk 支持模式匹配，若没有给出模式，则匹配输入文件中的所有行。如果成对出现，pattern 和 action 是相依赖的。awk 判断是否满足 pattern 模式，如果满足，则执行后面的 action；不满足，则什么也不干。比如如下这个命令：

```
awk '/set/ {print "you and me!!"}' .bashrc          #输入此行，下面是输出结果
you and me!!
you and me!!
you and me!!
```

上面命令的意思是在.bashre 中寻找 set，如果找到就输出一个“you and me!!”，结果在文中找到了 3 个 set，所以就输出了 3 个“you and me!!”。

2．awk 执行步骤

awk 执行命令时会反复进行下列四个步骤：

（1）自动从指定的数据文件中读取一个数据行；

（2）自动更新（Update）相关的内建变量值，如 NF、NR、$0 等的值均会被更新；

（3）依次执行程序中所有的 Pattern{Actions}指令；

（4）当执行完程序中所有 Pattern {Actions}时，若数据文件中还有未读取的数据，则跳转到步骤 1。

3．awk 的变量

awk 中有些默认的变量，使用这些变量会让处理文本变得更加方便，这些默认的变量如表 6.6 所示。

表 6.6　awk 中默认变量

变 量 名	含　义
$0	当前读取的行的内容
$1	$0 中第一列文本的值（以空格或 TAB 隔开的一项称为一列
$2	$0 中第二列文本的值，依此类推，$3，$4，$5…
NF （Number of Fields）	$0 中包含文本的列数
NR （Number of Records）	当前已读入数据行的数目
FILENAMEawk	当前处理的文本的名字
ARGC	命令行变元个数
ARGV	命令行变元数组
FILENAME	当前输入文件名
FNR	当前文件中的记录号
FS	输入域分隔符，默认为一个空格
RS	输入记录分隔符
OFS	输出列分隔符
ORS	输出记录分隔符

4．awk 用法示例

我们用一个文本的例子来说明 awk 的用法，例程文本文件名为 uandme，内容如下：

```
you1 and1 me1
you2 and2 me2
you3 and3 me3
you4 and4 me4
```

（1）使用这个命令：“awk '{print NR,NF,$1,$NF}' uandme”，这个命令的功能是显示文件 uandme 的当前记录号、列数和每一行的第一个和最后一个列。输出如下：

```
1 3 you1 me1
2 3 you2 me2
3 3 you3 me3
4 3 you4 me4
```

（2）再使用一下模式匹配的功能，输入如下命令：

```
awk '$1 * $2 !=100 {print "error"}' uandme
```

功能是如果当前行的第一列和第二列内容的乘积不为 100，则输出 error，命令运行后，应该可以看到四行 error。

（3）awk '/you2/ {print $1，$2 + 10}' uandme：在当前行中寻找 you2，如果找到，则输出第

1 列，并给第 2 列加 10。

（4）awk '$1 == "you2" {$3 = "lover" ; print}' uandme：在当前行中寻找“you2”，找到的话，将第 3 列变量替换为“lover”，然后输出。

（5）awk '/you/ {count++;} END {print "you was found "count" times"}' uandme：可以在命令行中使用 END 关键字，意思是执行完前面的所有命令，最后执行这行命令。

5．几个其他例程

awk 还可以实现许多复杂的功能，其他一些使用例程如下所示。

- awk '{ print FILENAME，$0 }' file1 file2 file3>fileall：把 file1、file2、file3 的文件内容全部写到 fileall 中，格式为输出文件并前置文件名。
- awk ' $1!=previous { close（previous）; previous=$1 }{print substr（$0，index（$0，" "）+1）>$1}' fileall：把合并后的文件重新分拆为 3 个文件，并与原文件一致。
- awk 'BEGIN {"date"|getline d; print d}'：通过管道把 date 的执行结果送给 getline，并赋给变量 d，然后输出。
- last |awk '{print $1}'|awk '{ for（i = 1; i<= NF; i++） freq[$i]++ } END { for（word in freq） printf "%s\t%d\n"， word， freq[word]}'：查询用户登录次数。

6．awk 的一些资料

awk 号称是编程语言，所以其有着编程语言的特性，如支持操作符、拥有函数等。awk 支持的操作符比较多，可以进行许多类型的比较，如测试大小等。表 6.7 列出了 awk 的操作符及允许的测试操作，以供查阅。

表 6.7　awk 的操作符及允许的测试操作

操 作 符	含　义
x==y	x 等于 y?
x!=y	x 不等于 y?
x>y	x 大于 y?
x>=y	x 大于或等于 y?
x<=y	x 小于或等于 y?
x～re	x 匹配正则表达式 re?
x!～re	x 不匹配正则表达式 re?

awk 的操作符（按优先级从低到高排列）：=、+=、-=、*=、/=、%=、||、&&、>>=、<<=、==、!=、～、!～、xy（字符串连结，'x"y'变成"xy"）、+、-、*、/、%、++、--。

awk 的函数非常丰富，包括数学函数、字符串处理函数等。表 6.8 列出了常见的函数及其功能，以供查阅。

表 6.8　awk 的函数及其功能

函　数	功　能
int(x)	对 x 取整
rand	取 0 到 1 之间的随机数
srand(x)	设置 x 为 rand 的新输入值
cos(x)	给出 x 的余弦值

续表

函　数	功　能
sin(x)	给出 x 的正弦值
atan2(x，y)	给出 y/x 的正切值
exp(x)	给出 e 的 x 次幂
log(x)	给出 x 的常用对数值（基为 e）
sqrt(x)	给出 x 的正平方根值
exit(x)	结束 awk 程序，若有 x 值，则返回 x，否则返回 0
index(s，t)	返回 t 在 s 中的第一个开始位置，如 t 不是 s 的子串，则返回 0
length(x)	求 x 的长度（字符个数）
substr(s，x，y)	在字符串 s 中取得从 x 个字符开始的长度为 y 的子字符串

6.4.5　sed 流编辑器

sed 是一个“编辑器”，但与其他大多数编辑器不同的是，其既不面向屏幕，又是非交互式的。这意味着使用时必须将要 sed 命令插入到命令行或脚本中。sed 接收这些命令并自动执行。就像水流过一样，因此而得名流编辑器。其非常适合于执行重复的编辑，这种重复编辑如果由人工完成将花费大量的时间。

而在进行文本文件处理时，如果不断地重复某些编辑动作，则可用 sed 自动一次执行这些编辑动作。当文件需要许多不同编辑动作时，则 sed 一次可执行那些不同的编辑动作。其参数可能和一次性使用一个简单的操作所需的参数一样有限，或者和一个具有成千上万行要进行编辑修改的脚本文件一样复杂，例如 sed 能一次执行完将文件中所有空白行删除、替换字符串、并将使用者输入的文字添加在文件的末尾等不同的编辑动作。

sed 的工作方式与前面介绍的 awk 基本一致，均是读入一行文本，然后执行为该行指定的所有操作，并在完成请求的修改之后将该行放回到内存中，以将其转储至终端。完成了这一行上的所有操作之后，它读取文件的下一行，然后重复该过程直到它完成该文件。如同前面所提到的，默认输出是将每一行的内容输出到屏幕上，当然，用重定向符可将其重定向到其他位置。流编辑器有两个重要特点：第一，输出可以被重定向到另一文件中，以保存变化；第二，不直接修改源文件。sed 的语法如下：

```
sed [options] '{command}' [filename]
```

options 指定 sed 执行的方式，可取如下值：

- -e：是编辑模式，在命令行上运行。
- -f：是文件模式，读取 sed 命令文件，然后运行。
- command：指定对文本进行处理的方式，其实就是指定一个 sed 可以识别的命令。
- filename：要处理的文件，可以指定多个文件。

1．sed 命令简介

先从具体的例子上来看 sed 是如何工作的。下面是一个替换文本的例子。

```
sed 's/you/westnorth/g' uandme
```

这条命令的作用是将 uandme 文件中的 you 替换为 westnorth。“s/you/westnorth/g”是语法

中的 command 部分，这部分是使用 sed 的关键所在。里面的“s”是编辑命令，意思是用第二个“/”后面的内容替换第一个“/”后面的内容，本例中是用“westnorth”替换“you”。“s”是 sed 的命令，而类似的命令在 sed 里很多，将其列于表 6.9，以便查看。

表 6.9　sed 命令及其功能

sed 命令	功　能
: label	为 b 与 t 命令创建标签
#	创建注释
{ }	标记一个命令块
=	显示当前行号
a\	添加文字，\是用来连接行的
b label	跳转到由:label 命令标记的标签处
c\	替换当前选中的行
d	删除数据
l	显示当前行内容
i\	插入数据
n/N	读入/添加下一行数据
p	输出当前处理的文本内容
q	退出 sed
r	读取另一个文件
s	替换字符串
w	将当前处理的数据写入一个文件
y	翻译字符

2．sed 用法示例

下面用一些详细的例子来说明命令的用法。

- sed -n '3,8p' myfile：输出第 3 到第 8 行，也可以只指定一个数字表示只输出某一行。
- sed -n '$p' myfile：输出最后一行。
- sed '/you1/ a\ aremystar′　myfile：寻找包含 you1 的行，在其后面添加一行，内容为 aremystar，如果要寻找不包含 you1 的行，然后添加一行，只需在 a 前面加一个“!”，像这样：“sed '/you1/ !a\ aremystar′　myfile”。
- sed '/your2 i\ ILoveyou′　myfile：在含有 you2 的行前面插入一行，内容为 ILoveyou，这个在 I 前加一个“!”号，也可以实现在不包含“!”的行前插入一行的功能。
- sed '5 a\ the5thline′　myfile：在第 5 行后插入一新行，内容为 the5thline。
- sed G myfile：在每一行后面增加一空行。
- sed '/^$/d;G'：将原来的所有空行删除并在每一行后面增加一空行。这样在输出的文本中每一行后面将有且只有一空行。注意这里使用了“;”将两个命令隔开，“;”后面的语句功能与 sed G 是一样的。
- sed = myfile | sed 'N;s/\n/\t/'：将文件的每一行先左对齐，然后进行编号。
- sed '1d' myfile：删除第 1 行，也可以输入 3，8d 这样的数字指定删除的行的范围。
- sed 's/my/westnorth/g' myfile：将所有 my 替换成 westnorth，g 表明全局替换。可以使用

“&”字符，该字符表示要搜索的目标字符串，在本例中表示 my。如果这样使用：“sed 's/my/&westnorth/g’ myfile”，则将文本中的 my 字符串替换为 mywestnorth。

- sed 's/$'"/`echo \\\r`/" myfile：将 DOS 的换行符（CR/LF）转换为 UNIX 格式。
- sed '/sweat/s/flower/rose/g' myfile：只在行中出现字符串 sweat 的情况下将 flower 替换成 rose。
- sed '/sweat/!s/flower/rose/g' myfile：只在行中未出现字符串 sweat 的情况下将 flower 替换成 rose。注意，这里又出现了取反符号“!”。

3．sed 一些技巧

sed 也可实现系统其他命令，但其语法过于复杂。在下面列出来也只是演示 sed 的功能而已。“：”前面为字符为系统命令，后面代码为 sed 实现与命令相同功能而使用的代码，如第二行“cat -s filename”就与使用“sed '/./，/^$/!d' filename”功能一样。

- cat：sed '' 或 sed -n '1,$p' 或 sed '1,$!d'。
- cat -s filename：sed '/./,/^$/!d' filename。
- cp file1 file2：sed 'w file2' file1。
- grep patten：sed -n '/patten/p' or sed '/patten/!d'。
- head：sed -n '1,10p'。
- tac：sed -n '1! G;$p;h' or sed -n 'G;$p;h'。
- tail -1：sed -n '$p' or sed '$!d'。
- tr "\n" " " ：sed ':a;N;s/\n/ /;ba'。
- wc -l：sed -n '$='。
- xargs：sed ':a;N;s/\n/ /;ba' or sed -e ':a' -e '$!N;s/\n/ /;ta'。

6.4.6 grep 文本查找命令

grep 系列应该是 Linux 中使用频率最高的文本查找命令了，主要功能是在一个或多个文件中查找特定模式的字符串，如果该行有匹配的字符串，则输出整个行的内容。如果没有匹配的内容，则不输出任何内容；与其他流编辑器一样，不改动源文件。Linux 的 grep 家族包括 grep、egrep 和 fgrep、rgrep，egrep 与 fgrep 不过是 grep 的一种模式而已，grep 可以通过-G、-E、-F 命令行选项来使用 egrep 和 fgrep 的功能。grep 使用格式如下：

```
grep [参数] [匹配的字符串] [目录或文件名]
```

意思是在“目录文件”中使用“参数”查找内容与“匹配的字符串”匹配的文件。如：

```
grep 'myfile’ myfolder/*
```

意思是打开 myfolder 目录，然后依次打开该目录中所有文件，在文件里查找 myfile 字符串，然后将包含 myfile 的每一行列出来。这是没使用参数的情况，如使用了参数，会改变 grep 命令处理的方式。像其他命令一样，grep 的参数有许多项，将其部分常用参数列出如下，以供查阅。

- -n：同时显示匹配行上下的 n 行，如：grep -2 pattern filename 同时显示匹配行的上下 2 行。与-C（大写）参数功能相同。
- -A NUM、--after-context=NUM：输出匹配的行，并且还输出该行后的 NUM 行内容，这个参数适用于名词解释，查找到名词后，再看看其后对于该名词的解释。
- -a 或--text ：将二进制文件看作文本文件进行搜索。

- -b、--byte-offset：输出匹配行前面有多少个字符。
- -B NUM，--before-context=NUM：与 -A NUM 功能类似，但此参数是显示匹配行与之前的 NUM 行。
- -c、--count：只输出匹配的行数，不显示匹配的内容。若再加上-v，--invert-match 参数则显示不匹配的总行数。
- -d ACTION、--directories=ACTION：若输入的名称是一个目录，使用 ACTION 去处理该目录。ACTION 默认值是 read（读取）；也可以将 ACTION 设置为 skip（忽略），则 grep 忽略所有目录。
- -E、--extended-regexp：采用正则表达式来处理匹配字符串。不用-E 参数的 grep 只支持普通正则表达式。将这些正则表达式元字符?、+、{、|、（、）视为普通字符，而失去其特殊意义。要使用这些元字符，须在其前加“\”，如\?、\+、\{、\|、\（、\）。加了-E 参数后，就可直接使用这些元字符。还可使用+匹配一个或多个先前的字符、?匹配零个或多个先前的字符、a|b|c 匹配 a 或 b 或 c，以及（、）分组符号。
- -r、--recursive：递归读取目录下所有目录及文件，与-d recsuse 参数功能相同。
- -s、--no-messages：不显示错误信息。
- -f File、--file=File：从文件中提取查找模式，如某些查找模式表达式较长且使用频繁，可将其写入一个文件，然后由本参数从文件中读取。
- -H、--with-filename：显示有匹配内容的文件名。
- -i、--ignore-case：匹配字符串时忽略大小写差别。默认情况区分大小写。
- -l、--files-with-matches：只列出匹配的文件名。
- -L、--files-without-match：输出不匹配模板的文件清单。
- -n、--line-number：在匹配的行前面输出行号。
- -q、--quiet：取消显示，只返回退出状态。
- -s、--silent：不显示关于不存在或者无法读取文件的错误信息。
- -v、--revert-match：反检索，只显示不匹配的行。
- -w、--word-regexp：把表达式作为一个单词搜索，完全匹配该单词的行才会被列出。如“grep -w magic*”只匹配'magic’，而不匹配'magical’。
- -x、--line-regexp：将搜寻样式视为一行去搜寻，完全符合该"行"的行才会被列出。
- -c、--count：不显示符合样式行，只显示符合的总行数。

grep 通过返回一个状态值来说明搜索的结果，如果成功，返回 0；不成功则返回 1；搜索的文件不存在，则返回 2。

grep 可以利用正则表达式来进行模式匹配，搜索特定的字符串，如下所示。

- ls -l | grep '^my'：只显示以 my 开头的文件或目录名（其中，文件和目录名由 ls 通过管道传送过来。
- grep 'end$' dict*：查找以 dict 开头的文件中以 end 结束的行。

6.4.7　显示可执行文件详情

需要查看的文件显然不是只有文本文件一种，有时，为了某些目的，可能需要查看可执行文件的内容，这时就可以使用本节介绍的命令了。

1．显示文件格式信息命令 file

file 命令的功能是显示文件的格式信息。其使用方法如下：

```
file 参数 testfile
```

然后就可得到 testfile 的文件类型，其可判别出文件是纯文本的代码，还是可执行程序，或者是一个链接文件等信息。下面以一些实际的例子来说明一些参数的用法：

示例 1：检测文件类型，简单使用 file 命令后面跟要检测的文件名，可以反映文件的类型。file 命令的输出有许多种，使用 file 检测了四种文件的输出内容如下：

```
earthfans@mylaptop:~$ file /usr/src/linux-2.6.27/init/main.c
/usr/src/linux-2.6.27/init/main.c: ASCII C program text
#检测表明 main.c 文件是一个纯文本的 C 程序代码
earthfans@mylaptop:~$ file /dev/sda
/dev/sda: block special                 #sda 是一个块设备
earthfans@mylaptop:~$ file tmp.tar.gz
tmp.tar.gz: gzip compressed data, from Unix, last modified: Sun Dec 14
07:50:21 2008
#tmp.tar.gz 是一个压缩文件
earthfans@mylaptop:~$ file /bin/ls
/bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for
GNU/Linux 2.6.8, dynamically linked (uses shared libs), stripped  #ls 是一个
elf 可执行程序
```

示例 2：file 还可检测分区信息，命令使用方法如下：

```
earthfans@mylaptop:~$ sudo file -s /dev/sda{1,2,3,4,5,6,7,8,9,10}
#依次显示系统中所有分区的信息，注意该命令要使用 root 权限
#如下输出显示了当前系统中的分区信息，这个依据不同的系统有不同的输出
/dev/sda1:  x86 boot sector
/dev/sda2:  x86 boot sector; partition 1: ID=0x7, starthead 1, startsector
63, 20482812 sectors; partition 2: ID=0x5, starthead 254, startsector 20482875,
257040 sectors
/dev/sda3:  ERROR: cannot open '/dev/sda3' (No such file or directory)
#这里显示错误，是因为 sda3 分区并不存在。sda4 显示错误的也是该原因
/dev/sda4:  ERROR: cannot open '/dev/sda4' (No such file or directory)
/dev/sda5:  x86 boot sector
/dev/sda6:  Linux rev 1.0 ext3 filesystem data (needs journal recovery)
#sda6 是 Linux 系统 ext3 分区格式，下面的输出可以同样解读
/dev/sda7:  Linux rev 1.0 ext3 filesystem data (needs journal recovery)
(large files)
/dev/sda8:  Linux/i386 swap file (new style) 1 (4K pages) size 124494 pages
#sda8 是交换分区格式
/dev/sda9:  Linux rev 1.0 ext3 filesystem data (needs journal recovery)
(large files)
/dev/sda10: ERROR: cannot open '/dev/sda10' (No such file or directory)
```

示例 3：可以使用 i 参数来判断文件的编码信息，使用方式如下所示：

```
earthfans@mylaptop:~$ file -i /usr/src/linux-2.6.27/init/main.c
/usr/src/linux-2.6.27/init/main.c: text/x-c charset=us-ascii   #输出显示main.c文件是us-ascii文件
earthfans@mylaptop:~$ file -i /dev/sda1
/dev/sda1: application/x-not-regular-file  #输出显示sda1分区并非普通文件
```

示例 4：使用 z 参数解读压缩文件的信息，如下所示：

```
earthfans@mylaptop:~$ file -z tmp.tar.gz        #使用z参数解读压缩文件
tmp.tar.gz: POSIX tar archive (GNU) (gzip compressed data, from UNIX, last modified: Sun Dec 14 07:50:21 2008)
#输出压缩文件的格式格式为gzip，在UNIX系统中进行压缩，最后一次修改时间是2008年11月14日
```


2．显示二进制文件的附加信息命令 objdump

objdump 是以一种可阅读的格式让你更多地了解二进制文件可能带有的附加信息。这个命令对于普通读者的意义可能不是很大，但有时也会用得上。比如有时一个软件的依赖是否满足或缺少哪些，则可以用这个命令查看软件的依赖详情。学习汇编语言时也可以将程序汇编成语言代码查看学习。不过要注意 objdump 操作的对象是可执行文件或类库，普通文本文件不能使用此命令。

objdump 的功能极其强大，使用很灵活，但参数也比较多。objdump 使用格式如下：

```
objdump[参数]objfile
```

参数的意义如下。

- --archive-headers、-a：显示文件库的成员信息。简单地说，这个参数可以显示文件类型。
- --disassemble-all、-D：反汇编二进制文件。
- -EB、-EL、--endian={big|little}：这个选项将影响反汇编出来的指令。little-endian 就是高位在高地址，x86 都是这种情况。
- --line-numbers、-l：用文件名和行号标注相应的目标代码，仅仅和-d、-D 或者-r 一起使用-ld 和使用-d 的区别不是很大，在源码级调试的时候有用，要求编译时使用-g 之类的调试编译选项。
- --architecture=machine、-mmachine：指定反汇编目标文件的格式，可以用-i 选项列出这里能够指定的格式列表。
- --reloc、-r：显示文件的重定位入口。如果和-d 或者-D 一起使用，重定位部分以反汇编后的格式显示出来。
- --dynamic-reloc、-R：显示文件的动态重定位入口，仅仅对于动态目标文件有意义，比如某些共享库。
- --full-contents、-s：显示指定 section 的完整内容。
- --source、-S：尽可能反汇编出源代码，尤其当编译的时候指定了-g 这种调试参数时，效果比较明显。隐含了-d 参数。
- --syms、-t：显示文件的符号表入口，类似于 nm-s 提供的信息。
- --dynamic-syms、-T：显示文件的动态符号表入口，仅仅对动态目标文件有意义，比如某些共享库。它显示的信息类似于 nm-D|--dynamic 显示的信息。
- --all-headers、-x：显示所有可用的头信息，包括符号表、重定位入口。-x 等价于-a、-f、

-h、-r、-t 同时指定。

示例 1：反汇编 open.o 类库，并将输出重定向至 open.s。使用如下：

```
objdump -D open.o >open.s
```

查看 open.s，其内容可能类似于这样：

```
open.o:     file format elf32-i386
Disassembly of section .text:
00000000 <csopen>:
   0:   55                  push   %ebp
   1:   89 e5               mov    %esp,%ebp
   3:   57                  push   %edi
   4:   56                  push   %esi
   5:   53                  push   %ebx
   6:   83 ec 3c            sub    $0x3c,%esp
   9:   8b 75 08            mov    0x8(%ebp),%esi
   c:   83 3d 00 00 00 00 00    cmpl   $0x0,0x0
……
```

这就是反编译出的汇编语言文件，如果有兴趣学习汇编的朋友可以经常使用此功能查看某些软件的汇编语言源码，应该会有收获。

示例 2：查询 os-apg.o 的符号表（相当于查看 os-apg.o 的调用函数列表），使用如下命令：

```
objdump -t os-agp.o
```

得到的部分输出可能类似于这样：

```
0000070 g     F .text    00000035 KernFreeAGPPages
00000000        *UND*    00000000 rm_free_agp_bitmap
00000000        *UND*    00000000 agp_unbind_memory
00000000        *UND*    00000000 agp_free_memory
000000b0 g     F .text   00000110 KernAllocAGPPages
00000000        *UND*    00000000 rm_alloc_agp_bitmap
```

说明 os-apg.o 程序中调用了 agp_unbind_memory、KernAllocAGPPages 等函数。

3．显示程序运行中的系统调用和信号的命令 strace

strace 命令是一种强大的工具，它能够显示所有由用户程序发出的系统调用、显示这些调用的参数并返回符号形式的值。可以从内核接收消息。其用法如下：

```
strace 参数 命令
```

常用参数：

- f：显示进程信息，同时也显示子进程的信息。
- o：将输出指定到一个文件中，而不是默认的屏幕输出。
- p：显示一个特定的进程的信息，进程号由本参数指定。
- e execve：只记录像 execve 这类系统调用。
- c：统计程序运行时间，及相关的系统调用统计信息。

- -e trace=file：只显示以指定文件名做参数的所有系统调用。
- -e trace=network：只显示和网络相关的系统调用。
- xx ：以 16 进制字符串格式显示所有字节。

如使用 strace 检测 ls 命令所执行的系统调用及信号，运行如下命令即可：

```
earthfans@mylaptop:~$ strace ls          #检测 ls 的系统调用
#下面是输出信息
execve("/bin/ls", ["ls"], [/* 42 vars */]) = 0
brk(0)                                  = 0x8573000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
mmap2(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)
= 0xb7f62000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=64229, ...}) = 0
mmap2(NULL, 64229, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7f52000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/tls/i686/cmov/librt.so.1", O_RDONLY) = 3
read(3,  "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0`\31\0\000"...,
512) = 512
…
```

前述命令显示了执行“ls”命令时所使用的一系列系统调用信息，有兴趣的读者可以分析一下其具体实现。

6.5　文件操作基本命令

文件操作包括对文件的处理，如移动、复制、创建、删除和更改权限等操作。在编写脚本文件、或在网络远程登录时，熟悉命令行的文件操作无疑具有很强的实用性和灵活性。在熟悉命令操作时，最重要的无疑是练习，书中给出基本的功能表和基本技巧，阅读时应充分发挥想象力，结合工作和生活中的需要对各种命令详加练习。

6.5.1　显示文件信息命令 ls

ls 命令是 Linux 中极常用的一条命令，功能是查看当前目录中的文件信息。如果不加任何参数，则显示当前目录中的文件名。简单地在 Shell 中输入 ls，则将当前文件名以各种不同颜色依次显示在屏幕上。

1．命令输出内容解读

使用 ls -l 命令来查看当前目录，会得到类似如下的输出：

```
westnorth@16:22 ~$ ls -lh
total 420M
-rw-r--r--  1 westnorth westnorth 378M 2008-10-02 21:05 backup.tar.gz
drwxr-xr-x  2 westnorth westnorth 4.0K 2008-09-30 03:07 bin
```

```
    drwxr-xr-x  2 westnorth westnorth 4.0K 2008-08-02 13:07 bochs
    lrwxrwxrwx   1  westnorth  westnorth     17  2008-09-28  21:01  mybook  ->
Documents/mybook/
    ……
```

下面我们来分析一下从输出中可以得到的信息。

第 1 行 total 420M 意思是目录中所有文件所占的空间为 420MB。

第 2 行起是文件信息，每行都有 8 列信息（下面解释第 1 行的每一列）：

- 第 1 列 drwxr-xr-x：总共是 10 位，告诉我们该文件的类型及权限，其中第 1 位是文件类型，2～10 位是文件权限，关于文件类型和文件权限，接下来再列表加以说明。
- 第 2 列 1：表示该文件的硬链接数，硬链接的概念在第 7 章会详细说明。
- 第 3 列 westnorth：表示该文件的属主（即谁拥有该文件所有权）。
- 第 4 列 westnorth：表示该文件的属组（即该文件属于哪一用户组）。
- 第 5 列 378M:表示该文件的大小。
- 第 6 列 2008-10-02：表示文件更新的日期。
- 第 7 列 21:05：表示文件更新的时间。
- 第 8 列 backup.tar.gz：文件名。

下面来介绍一下文件类型，文件类型共有 7 种，各用一个小写字母来表示。

- b：表示块设备（如硬盘、U 盘）。
- c：表示字符设备（如键盘）。
- d：表示目录。
- l：表示符号链接（指向另一个文件）。
- p：表示命名管道文件。
- s：表示套接字文件。
- -：表示不同文件，即不属于上述几种类型的文件。

文件权限有 4 种，用 rwx 来表示：

- r：可读（read）。
- w：可写（write）。
- x：可执行（exeute）。
- -：没有以上的权限。

查上表就知道前面的输出只有第一个文件是系统不识别的文件类型，第二、三个文件均为目录，最后一行的文件是一个符号链接，实际指向一个目录，地址是 Documents 目录中的 mybook 目录。

2．ls 参数

看懂了 ls 的输出信息，再来看看 ls 的使用，ls 使用很简单，类似于这样：

```
ls  [参数] [name...]
```

参数表的意义如下。

- -a：显示所有文件及目录（ls 默认不显示开头为"."的文件和目录）。
- -A：同-a，但不列出“.”（目前目录）及“..”（父目录）。
- -d：仅仅显示指定目录信息。目录和文件一样处理，这在当使用-l 标志获取目录状态时非常有用。

- -l：显示文件类型、权限、拥有者、文件大小等详细信息。
- -i：显示文件的 i 节点号。
- -r：将文件以相反次序显示（默认按英文字母次序显示）。
- -t：将文件按创建时间的先后次序列出。
- -F：在列出的文件名称后加一识别符号；例如可执行档则加“*”，目录则加“/”，如果为 FIFO，则放置一个|（管道）符号，如果是符号链接，则放置一个@，以便辨认。
- -R：递归显示文件，也就是说显示目录中的目录。
- -d：显示目录名，而不是显示目录中文件。
- -b：以八进制（\nnn）表示法显示非打印字符。
- -E：列出文件的保留空间、固定范围大小和范围分配标志信息。使用该标志时，必须指定-l 标志。
- -m：使用流输出格 s 式（逗号分隔系列）。
- -n：-n 标志除显示用户和组 ID 而不是用户和组名称以外，它显示和-l 标志同样的信息。
- -N：当确定文件状态时，不跟在符号链接之后。
- -p：如果文件是目录，在每个文件名后面放置一个斜杠。当将 ls 命令的输出通过管道传递到 pr 命令时，这很有用。
- -q：将文件名中的非打印字符显示成?（问号）。
- -s：按照千字节（包括间接块）给出每一项的大小。
- -u：使用最近访问时间，而不是最近修改时间用以排序（当使用-t 标志）或者显示（当使用-l 标志）。如果不和-t 或-l 标志或两者一起使用，则该标志无效。
- -x：以多列格式横向排序输出。
- -1：（数字 1）强制输出为每行一项格式。当输出没有定向到终端时，此为默认值。

3．用法示例

下面给出几个范例，ls 的使用并无特别的技巧而言，可以将上面介绍的参数逐个加入 ls 中查看输出效果，以加深对命令的理解：

- ls -ltr：列出目前目录中所有文件，显示详细信息，按创建的先后顺序排列，并按照逆序排列。
- ls -lRAF：递归显示所有子目录，显示隐藏文档，同时给文件名称后加识别符。
- ls -lF /bin/g* ：显示/bin 目录中以 g 开头的所有文件的详细信息，并给文件加上识别符。

6.5.2　文件移动、复制、删除命令

文件操作是一个操作系统的最为基本的功能。一般包括了文件的新建、复制、删除、移动等操作。不过，一般文件的新建工作均在相应的程序中进行，如新建文本文件一般在字处理程序中进行等。所以本节主要介绍文件的移动、复制、删除等命令。

1．文件移动及重命名命令 mv

mv 可以用来将文件或目录移动到另一个位置，移动后原始位置的文件或目录就不存在了。也可以使用该命令将一个文件或目录进行重命名，其使用方法如下：

```
mv 参数 源文件 目标文件
```

参数可取下列值。

- -f：强制移动，也就是说在覆盖现有文件之前不提示。
- -i：移动文件或目录前先进行提示。若回答“y”，则继续移动，否则取消移动。
- -r：递归处理目录中的所有文件及子目录。

用法示例：

- mv filea fileb：这是一个使用 mv 对文件重命名的例子，是将文件 filea 重命名为 fileb。如果文件 fileb 已存在，则其原有内容将由 filea 的内容替换；
- mv folder1 folder2：重命名目录。其中 folder1、folder2 均是目录名，如目录 folder2 已经存在，则执行此命令后，folder2 的原有内容被 folder1 的内容所取代；
- mv file1 folder/file2：将文件 file1 移动至 folder 目录下，并更名为 file2；
- mv a* folder：将当前目录以 a 开头的所有文件移动到 folder 目录中，可以在 mv 中使用其他的正则表达式来简化操作。

2．复制文件命令 cp

cp 可以文件或目录复制到另一个位置。复制完成后，原始位置的文件是存在的，这是其与 mv 命令的最大不同之处。cp 的用法如下：

```
cp [选项] 源文件 目标文件
```

选项可以取如下值。

- -f：如果目标文件存在而复制工作不能继续时，将存在的目标文件删除保证复制工作的顺利进行。
- -h：强制 cp 命令复制符号链接。不指定，系统默认是跟随符号链接，也就是将文件复制到符号链接指向处。
- -i：在覆盖前提示，须确认后才会进行覆盖。
- -p：复制文件时将文件的时间信息及用户、组信息等信息一并复制。如果用户标识或组标识不能复制，则文件许可位 S_ISUID 和 S_ISGID 被清除。
- -u、--update：只在源文件比目的地文件新，或目的地文件不存在时才进行复制。
- -r：递归复制，一般用来复制目录。
- -R：复制源文件层次结构到目标目录。特殊文件类型，如先入先出（FIFO）文件和块及字符设备将被重新创建而非复制。符号链接也被跟随，除非指定 -h 标志。（-R 标志优先于 -r 标志。）
- -s、--symbolic-link ：只创建符号链接而不是复制文件。

cp 使用比较简单，仅列举几个典型的例子来说明其具体操作：

- cp file file.backup：将文件在当前目录复制一份，功能是备份一个文件，在修改系统配置时常用；
- cp file folder1/folder2：将 file 复制到 folder1 目录下的 folder2 目录中；
- cp -r folder1/folder2 folder3/folder4：将 folder2 目录的目录结构及所有文件复制到 folder4 目录；
- cp -p myfolder backfolder：备份时应该使用-p 参数，这样备份的文件就还属于原来的组或用户。不然用 root 身份备份的文件，其属主是 root，很多时候普通用户使用会出问题。

3．删除文件命令 rm

rm 用来删除文件及目录，其使用方式如下：

```
rm [options] name...
```

options 可取如下参数。

- -i：删除前逐一询问确认。
- -f：即使原文件是只读文件也强制进行删除，而不进行询问。
- -r：递归删除目录及各子目录中的文件。

用法示例：

- rm file：删除文件。
- rm -r folder：删除目录。
- rm -ir folder：删除目录，不过在删除前先进行确认。

文件经过 rm 删除后，使用普通的方法是无法恢复的，所以这个命令要慎用。尤其在输入时要特别小心。比如要删除当前目录中的 file 目录，输入“rm -rf . /file”，注意在“.”和“/”之间多输入了一个空格，这样就会删除主目录中的所有文件，而且删除了的文件不易恢复。想象一下删除了主目录中所有文件会发生什么事。预防措施一般是用别名功能强制加上-i，如在.bashrc 中加上一句：alias rm='rm -i’。不过正如前面所说的，这样做的问题是如果删除文件很多，逐个确认也是一件很麻烦的事情。

6.5.3　创建文件链接、分割文件

本节介绍了文件链接的基本概念，符号链接与硬链接的区别，以及创建文件链接的命令 ln、分割文件的命令 split 和更改文件信息的命令 touch。

1．创建文件链接命令 ln

文件链接相当于文件的一个快捷方式，使用 ln 可以创建文件的符号链接与硬链接文件。命令的使用方式如下：

```
ln [options] source dist,
```

其中 option 的可选参数如下。

- -f：链接时先将与 dist 同名的文件删除。
- -d：允许系统管理者硬链接自己的目录。
- -I：在删除与 dist 同名的文件时先进行询问。
- -n：在进行软链接时，将 dist 视为一般的文件。
- -s：创建一个软链接，也称为符号链接（symbolic link）。
- -v：在链接之前显示其文件名。
- -b：将在链接时会被覆盖或删除的文件进行备份。
- -S SUFFIX：将备份的文件都加上 SUFFIX 的后缀。
- -V METHOD：指定备份的方式。
- --help：显示帮助信息。
- --version：显示版本信息。

使用很简单，“ln myfile yourfile”产生一个硬链接，“ln -s myfile yourfile”产生一个符号链接。符号链接与硬链接可能许多人还是很陌生，在此简单地介绍一下。硬链结的意思是一个文件可以有多个名称，而符号链结则是产生一个特殊的文件，该文件的内容是一个字符串，

字符串的内容是另一个目录或文件的目录，凡是对符号链接文件的操作都通过链接被传递到目标文件。

符号链接与硬链接的不同在于符号链接可以连接到目录，而硬链接不行。符号链接是一个指向目标文件的连接；而硬链接对于系统来说，只不过一个文件的另一个名字；在效率上，硬链接稍微好于符号链接，而符号链接灵活性更强。用两个例子来说明两种连接的不同之处，假设在当前目录有两个文件 pet1 与 food：

（1）建立同一个文件的硬链接与符号链接文件，如下所示：

```
ln -s pet1 dian                    #建立符号链接文件
ln food nice                       #建立硬链接文件
```

分别建立了这两个文件的链接，其中“dian”是符号链接，而“nice”是硬链接。

（2）使用任何文本查看命令分别查看两个文件的内容，应该没有任何问题。

（3）现在删除这两个源文件，执行如下命令：

```
rm pet food
```

（4）再查看两个链接的内容，使用任何文本查看命令，如 cat，如下所示：

```
cat dian                           #查看符号链接文件内容
cat nice                           #查看硬链接文件内容
```

前者会报错，后者还能正常显示。因为符号链接指向的目标文件不存在，所以无法显示内容。而硬链接实际上是创建了一个原文件的副本，所以原文件删除后，副本仍然存在，还可以正常显示。这个例子很好地说明了符号链接和硬链接。

2．分割文件命令 split

split 可以将文件分割为指定大小的若干块。其使用方式如下：

```
split [参数][file[备份文件名前缀]]
```

默认情况下，split 命令读取 file，然后按 1000 行为单位将其分割成一组输出文件。第一个输出文件名由指定前缀（默认值为 x）和 aa 后缀组合构成，第二个文件名由前缀和 ab 后缀组合构成，如此按字典顺序一直到 zz（最多 676 个文件）。如果未指定输入文件或如果指定为“-”（减号），那么 split 命令从标准输入读取文件。命令的“参数”可取下列值。

- -b、--bytes=n：指定输出文件的大小。
- -C、--line-bytes=n：指定输出文件中每行的最大长度。
- -l、--lines=n：指定每个输出文件的最大行数。
- -a：指定输出文件名的前缀长度，默认是 2。
- -3（或其他任意数值）：与 l 参数的功能相同。

用几个例子来说明参数的用法。

- split mylog：将 mylog 文件分割为 1000 行一个的若干个小文件。
- split -b 1m mylog：将 mylog 文件分割为 1m 大一个的若干个小文件。
- split -l 500 mylog：将 mylog 分割为 500 行一个的若干个小文件。
- split -a 3 myhuge：将文件的后缀长度指定为 3，形式类似于“aaa，aab…”，这样最大可以将文件分割为 26*26*26＝17576 份。

3．更改文件时间信息 touch

touch 可更改文件的时间信息，如文件的访问和修改时间等，其使用方式是：

```
touch [参数] file1 [file2 ...]
```

命令的参数可取下列值。

- f：强制使用 touch 来修改文件时间信息。
- a：只改变文件的读取时间记录。
- m：只改变文件的修改时间记录。
- c：假如目的文件不存在，不会建立新的文件，与--no-create 的效果一样。
- r：使用另一个文件的时间信息，与--file 的效果一样。
- d：设定时间与日期，可以使用各种不同的格式。
- t：使用指定时间而不是当前时间。Time 变量以十进制形式[[CC]YY]MMDDhhmm[.SS]指定，其中：CC 指定年份的前两位数字，YY 指定年份的后两位数字，MM 指定一年的哪一月（从 01 到 12），DD 指定一月的哪一天（从 01 到 31），hh 指定一天中的哪一小时（从 00 到 23），mm 指定一小时的哪一分钟（从 00 到 59），SS 指定一分钟的哪一秒（从 00 到 59）。

用几个例子来说明参数的用法。

- touch file：最简单的使用方式，不带任何参数使用 touch，则将文件的时候记录改为现在的时间。若文件不存在，系统会建立一个新的文件。同时这也是一种优雅的创建文件的方法。
- touch -r file2 file：用 file2 的时间信息来设置 file1 的时间信息。
- touch -t 10252050 file：设置 file 的时间为 10 月 25 日 20 时 50 分。

6.5.4　目录操作

Linux 的目录表示中有几个特定的字符，这些字符代表特定的目录，一些常用的目录指代字符如下所示。

- “/”：根目录，是一切目录的父目录，只有第一个字符是“/”才代表根目录，其余地方出现的“/”仅代表目录分隔符。如“/usr”代表根目录中的 usr 子目录，这里的“/”代表根目录。而“～/Document”中的“/”则仅是一个目录分隔符。
- “～”：主目录，是当前用户的主目录，一般是“/home/用户名”。
- “.”：当前工作目录。
- “..”：上级目录。

目录与子目录之间用“/”隔开，像这样“./myfolder/mymusic”。对于目录的操作应该是本章最简单的一部分，几个命令也是简单明了，而且参数也不多。

1．显示当前目录 pwd

这个命令比较简单，功能是输出当前目录。记忆这个命令的技巧是记住其全称（print work directory）“输出工作目录”。很多时候，需要知道当前目录，以便根据目录来进行相应的操作，这时就用得上 pwd 了。

pwd 的参数中比较常用的是-P，其作用是显示当前目录的绝对地址。下面通过一个示例介绍该参数的用法：

```
$mkdir folder1                          #建立一个目录。
$ln -s folder1 sym1                     #建立该目录的符号链接 sym1
$cd sym1                                #改变当前目录到符号链接的目录 sym1 中
$pwd
/home/westnorth/sym1                    #显示的是符号链接的目录
$pwd -P
/home/westnorth/folder1                 #加上参数 P 后，显示的是绝对地址
```

2．更改当前目录命令 cd

Linux 的一个哲学是就是简单就是美，所以最常用的几个命令均极其简单。比如这个改变目录的命令 cd，其用法很简单，以至于系统都没有给出 man 的帮助页，可以自己 mancd 试一下。使用格式为“cd 目标目录”，就可以改变目录到目标目录了。当然还有一些技巧，列在下面，供使用时参考。

- cd：改变当前目录为用户的主目录。
- cd ～：与上条命令一样，均是回到主目录。
- cd ..：到上一级目录，注意在 cd 和“..”之间有个空格。
- cd -：回到上次所在的目录。
- cd !$：把上一个命令的最后一个参数拿来当 cd 的参数使用。
- cd !^：把上一个命令的第一个参数拿来当 cd 的参数使用。

下面是一个简单的使用 cd 各种技巧改变目录的例子，可以作为练习的参考：

```
cd /usr/bin/                #更改至一个目录
cd ～/software              #更改至第二个目录
cd -                        #这条命令是回到上一个目录，这里会回到/usr/bin
tar xzf pac.tar.gz -C ～/software
cd !$                       #取上一个命令的最后一个参数，等于是执行 cd ～/software
```

3．创建目录命令 mkdir

创建一个目录可以使用 mkdir，其用法如下：

```
mkdir  [参数] dir-name
```

命令中各参数的含义如下。

- - m：对新建目录设置存取权限，也可以用 chmod 命令设置。
- - p：可以是一个路径名称。此时若路径中的某些目录尚不存在，加上此选项后，系统将自动建立好那些尚不存在的目录，即一次可以建立多个目录。

该命令要求创建目录的用户在当前目录中具有写权限，并且 dirname 不能是当前目录中已有的目录或文件名称。还是用几个例子来说明这些用法。

```
mkdir myfolder              #创建一个目录
mkdir -m a+rw folder1       #创建一个用户有读写权限的目录
mkdir -p this/is/myworld    #依次创建 this、is、my、world 几个目录
```

4．删除目录命令 rmdir

rmdir 是删除目录的命令，但其只能删除空目录，用法如下：

```
rmdir [-p] dirName
```

参数只有一个：

- -p：是当子目录被删除后使它也成为空目录，则顺便一并删除。

用几个例子来说明参数的用法。

```
rmdir left                  #将当前目录中名为 left 的子目录删除。
rmdir -p dad/child
#删除 dad 目录中的 child 子目录，如果删除 child 后，dad 也是空目录，则继续将 dad 目录删除。
```

5．显示树形目录结构 tree

tree 可以递归显示指定目录中文件与目录的列表图，如下所示。

```
westnorth@21:16 ～/Document$ tree
.
|-- How to Install Wikipedia on Your iPod | MakeUseOf.com.mht
|-- How to Install Wikipedia on Your iPod | MakeUseOf.com_files
|   |-- 468×60-c.png
|   |-- brand000.htm
|   |-- briansth.png
|   |-- briansti.png
|   |-- counter0.js
|   |-- date0000.gif
……
```

其他一些常用参数如下。

- d：只显示目录。
- f：显示完整路径。
- i：不显示缩进及前导符，如使用 f 参数时，可配合此参数使用。

作为参考，我们也提供了几个简单的例子，以供参考。

```
westnorth@21:20 ～/.wine$ tree -fid
#以不显示缩进及前导符，显示完整目录的方式显示当前树形目录
.
./dosdevices
./dosdevices/c: -> ../drive_c
./dosdevices/z: -> /
./drive_c
./drive_c/IEGD
./drive_c/IEGD/IEGD_8_0/jre/bin
./drive_c/IEGD/IEGD_8_0/jre/bin/jrockit
./drive_c/IEGD/IEGD_8_0/jre/lib/zi
./drive_c/IEGD/IEGD_8_0/jre/lib/zi/Africa
./drive_c/IEGD/IEGD_8_0/plugins
./drive_c/IEGD/IEGD_8_0/plugins/com.intel.iegd.drivers_8.0.0
```



tree 命令不是系统自带的命令，使用前需要运行 sudo apt-get install tree 进行安装。

6.5.5 文件查找命令 find

find 命令是直接读取磁盘文件，然后在其中寻找相应的文件。这种寻找方式比较慢，但该命令提供了丰富的功能，弥补了这点缺憾。find 命令可以根据权限、时间信息、大小等各种信息查找文件，对于特种需求总是可以满足的。找到文件后，find 命令还可调用其他 Shell 命令进行后继操作，这就给各种复杂功能的实现提供了方便。

1．命令格式

其完整使用格式如下：

```
find [路径] -参数 [-exec [Shell 命令{} \;] ]
```

几个参数的意义如下。

- 路径：是 find 命令所查找的目录路径。
- -exec：对匹配的文件执行该参数所给出的 Shell 命令。

参数表及其功能列于下。

- -name：按照文件名查找文件。
- -perm：按照文件权限来查找文件。
- -user：按照文件属主来查找文件。
- -group：按照文件所属组来查找文件。
- -mtime -n +n：按照文件的更改时间来查找文件，-n 表示 n 天以内，+n 表示 n 天前。
- -nogroup：查找无属组的文件。
- -nouser：查找无属主的文件。
- -type：查找某一类型的文件。

类型可为下列类型中的任一个。

```
b ：块设备文件
d ：目录
c ：字符设备文件
p ：管道文件
l ：符号链接文件
f ：普通文件
```

- -mount，-xdev：只检查和指定目录在同一个设备中的目录，避免列出其他设备的文件。
- -amin n ：在过去 n 分钟内被读取过。
- -anewer file：比文件 file 更晚被读取过。
- -atime n：在过去 n 天过读取过的文件。
- -cmin n ：在过去 n 分钟内被修改过的文件。
- -cnewer file：列出比文件 file 新的文件。
- -ctime n：在过去 n 天修改过的文件。
- -empty：空的文件-gid n or -group name：gid 是 n 或是 group 名称是 name。
- -ipath p，-path p：路径名称符合 p 的文件，使用 ipath 参数则会忽略大小写。
- -name name，-iname name：文件名称符合 name 的文件。iname 会忽略大小写。
- -size n：文件大小是 n，＋n 表示大于 n，－n 表示小于 n，n 表示等于 n。

2．用法示例

下面是几个使用 find 参数进行高级文件查找的示例。读者可参考前述参数，自行组合复杂的查找命令进行练习：

```
find . -perm 755              #查找当前目录及子目录下所有文件权限为 755 的文件
find ～ -user root            #在用户目录下查找文件属主为 root 的文件
find /home -nouser            #查找/home 目录下那些没有属主的文件
find -mtime -4                #在当前目录下查找 4 天内修改过的文件
find /usr/share -mtime +4     #在/usr/share 中查找 4 天前修改过的文件
Find /etc/my*  -nouser        #查找/etc 目录中以 my 开头的无属主的文件
```

3．对匹配的文件执行特殊操作

find 命令还有一个-exec 参数用来执行 Shell 命令。当找到匹配的文件后，可能还希望对其进行某种操作，如分割，压缩，这时就可以使用-exec 选项。这个使用有点复杂，不过要记住其使用格式：

```
find “查找条件” -exec  Shell 命令 {} \;
```

注意其后的命令“{}（空格）\;”，在大括号“{}”之前有一空格，之后也有一空格，但两个大括号之间没有空格。“\;”的目的是转义“;”字符（因为“;”是 Shell 的关键字，所以需要给其加“\”转义），“\”“;”之间不能有空格。这些格式必须严格遵守，如果漏写了任何一个符号，都会得到一个语法错误的提示。同时这些格式也是命令的一部分，不能省略。其实对搜索结果进行处理，理所当然的想法是使用管道将搜索结果传送至下一个 Shell 命令，但这个方式可以使用参数，使用更为灵活。来看几个例子就明白了。

- “find ～ -type f -mtime +80 -name "*.Backup" -exec od {} \;”在用户主目录下查找修改时间在 80 天的“. Backup”，然后将其显示出来。
- “find ～/Music -type f -name "wanted*" -exec cp {} ./myfavority \; ”：在主目录中的 Music 目录中寻找名字以 wanted 开头的文件，然后将其复制到当前目录的 myfavority 子目录中。
- “find ～/develop -name "*.h" -exec grep SET_TIME {} \;”在主目录的 develop 子目录中查找所有的 C 语言头文件，然后在文件中查找“SET_TIME”字符串。

6.5.6　快速查找命令及特殊查找命令

除了 find 命令外，系统还提供了几个其他的命令，以用于多种用途。如快速但简单的查找命令 locate，查找可执行程序的 which 命令等。

1．快速查找命令 locate

locate 是一个快速查找文件的命令，使用 locate 基本不用等待，马上就可以看到搜索结果。但不知为什么，这个命令似乎许多人都不知道，或者至少不常用。虽然本命令有些弱点，但如果仅仅是简单地根据名字信息查找文件，这条命令无疑是首选，因为其速度实在不是 find 可以比拟的。

locate 的弱点是查找结果不一定能反映文件的最新变动，因为其不直接读取磁盘信息，而是在一个记录了文件名字和目录的数据库进行查找。如果文件目录有了变动，而数据库还未及时更新，就可能会出现错误。更新数据库是由系统自动运行的。如果要自己更新文件数据库信息，可以用“sudo updatedb”来更新。locate 命令的使用很简单，如下：

```
locate [参数] 文件
```

解释一下参数：

- -b、--basename：只显示目录最底层与搜索文件名匹配的项目。
- -c、--count：显示共找到文件的数量而不是显示文件名。
- -i、--ignore-case：忽略文件名中的大小写。
- -q：安静模式，不会显示任何错误信息。
- -l：最多显示 n 个输出。
- -r：使用正则表达式作为寻找的条件。
- -d：指定数据库的名称。
- -S：（大写）显示数据库的统计信息。
- -0：使用 ASCII 符 NUL 作为文件之间的分隔符（默认是回车符）。

范例：

- locate file：寻找系统中所有以 file 为名字的文件，然后将其列出来。
- locate file | less：locate 命令当然支持管道操作符，它将文件列出来重定向到命令 less，这样做目的是方便进行翻页和查找等相关工作。
- locate -r history$：查找以 history 结尾的文件。
- locate -c westnorth：查找名字中包含 westnorth 的文件的数量。
- locate -cbi westnorth：统计最底层目录名或文件名中包含 westnorth（忽略大小写）的文件的数量。
- locate -S：数据库的相关信息。输出可能类似于：

```
Database /var/lib/mlocate/mlocate.db:
51,160 directories
645,913 files
50,161,423 bytes in file names
15,738,569 bytes used to store database
```

说明该数据库存储于/var/lib/mlocate 目录，名为 mlocte.db。存储了 51160 个目录、645913 个文件，文件、目录名共有 50M 字节的数据，但实际占用空间是 15M（说明数据库进行了压缩）。

2．which 查找程序文件

which 也是一个查找文件的程序，但其查找范围只是环境变量$PATH 中定义的目录，而且只查找可执行程序文件名，因此速度极快。使用很简单，格式如下：

```
which [程序名 ...]
```

如果找到该程序，就会列出程序的目录。比如，运行命令“which which”，查找“which”命令的位置，就会得到类似这样的输出：

```
/usr/bin/which
```

说明 which 程序在/usr/bin/目录。

3．whereis

是一个查找命令，但与普通的查找命令有点不同的是：其查找目的是找出与特定文件有关系（文件的可执行文件的位置，文件的源代码，文件的说明手册）的几种文件的位置。whereis

命令理解的文件类似于图 6.6，每个文件都有其对应的三部分，而 whereis 命令会找出与文件名相关或无关的文件。其运行方式是：

```
whereis [参数 1] [ 参数 2 目录 -f ] 文件
```

参数 1 指定要查找与文件有什么关系的文件的位置，共可取四个参数，分别如下。

- -b：搜索可执行程序的位置，与 which 相同。
- -m：搜索文件的手册。
- -s：搜索文件的源代码。
- -u：搜索其他文件。相当于一个“非”命令，使用这个参数则不寻找特定的部分，如-u -m 寻找文件的除了手册的其余部分。
- -B、-M 和-S：意义与小写的一样，但是这三个参数需要指定搜索的目录，而且这个目录必须是绝对路径（以“/”开始的目录）。
- -f：结束前面-M、-S 或-B 参数的目录字符串（也就是说，必须和前面的这三个参数或三者之一共同使用），开始搜索文件名字符串。

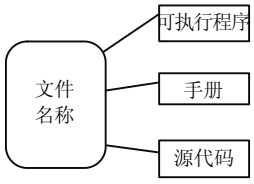


图 6.6　whereis 命令查找文件的 3 个部分

用法示例。

- whereis -u -M /usr/man/ -S /usr/src -f *：意思是查找当前目录中的所有文件（*可以匹配任何字符串），前提是在/usr/man 中没有手册（-u -M 参数产生这个功能），或者在/usr/src 目录中没有源代码（-u -S 参数产生这个功能，只用一个-u 就可以将所有参数取反）。
- whereis ifconfig ：查找 ifconfig 所有部分的位置，包括可执行程序所在的位置，手册的位置，及源代码的位置。

6.6　其他命令

Shell 命令极其繁多，本章前面介绍的只是常用的一些命令。还有其他的许多命令，可以完成许多实用的功能，本节将介绍其中部分命令。

1．系统休眠命令 sleep

该功能是让当前程序休眠一段时间，然后再执行指令。使用方式如下：

```
sleep num [单位]
```

num 是指时间长度，后面跟的是时间单位，可取值有：s 为秒，m 为分钟，h 为小时，d 为天数。

如“sleep 20 m”是指让程序休息 20 分钟后再继续执行其他指令。

2．获得程序运行时间 time

time 并不像其名字那样只能获得时间信息，还能获得程序运行时消耗的其他资源，如内存、IO、命令的信息等。其基本使用格式是：

```
time [参数] 命令 [命令的参数]
```

参数取值如下：

- -o、--output=FILE：指定将结果输出至名为 FILE 的文件中。
- -a、-append：与-a 共同使用，将输出添加至一个文件。
- -f FORMAT、--format=FORMAT：设置输出显示的格式，要查看 time 命令获得的其他信息，如程序消耗的内存、IO 等信息，就要在格式显示里设置，默认格式是这样的：
 real %e：程序运行时间。
 user %U：用户模式运行时间。
 sys %S：核心模式运行时间。

用户可以使用自定义格式来改变输出内容。“-f” 参数可以使用的格式有许多，但现在有些发行版的 tme 命令似乎有点问题，不能使用许多参数。就不具体介绍这些参数，如果读者手头的发行版可以正常使用 time 命令，请自行查阅 man。

- -p、-portability：这个选项会自动把显示格式设定成为：real %euser %Usys %S 这么做的目的是为了与 POSIX 规格相容。
- -v、-verbose：将所有程序使用过的资源都列出来，可以获得最丰富的信息。

例如，利用下面的指令就可以获得执行 ps -aux 命令的结果和所花费的系统资源：

```
time ps -aux
```

3．管理和设置日期时间命令 date

date 可以用来显示或设定系统的日期时间，使用很简单。如下所示：

```
date [参数]
```

如果不加参数，就显示系统当前日期（包括年、月、日、星期）与时间。如果对于时间显示有自己的需求，可以自定义显示格式，方法为一个加号后面跟几个参数，如“date +%d "today I'll be happy"”，输出可能类似于这样“17：I'll b happy”。

下面是一些自定义显示格式时可用的参数。

- %H：当前小时数。
- %M：当前分钟数。
- %S：当前秒数。
- %s：从 1970 年 1 月 1 日 00:00:00 UTC 到目前为止的秒数。
- %T：显示时：分：秒。
- %B：以当前语言设置（Locale）表示的月份。
- %A：以当前语言设置（Locale）表示的星期几。
- %d：当前日期。
- %D：显示年 / 月 / 日。
- %j：今天是一年中的第几天。
- %m：数字表示的月份。
- %W：本周是一年中的第几周。
- %Y：当前年份。
- -u：显示目前的格林尼治时间。

例子：

显示月份与日数“date +%B %d”，该命令的显示应该类似于这样“十二月 23”。

4．获得系统相关的时间信息 uptime

uptime 的使用很简单，没有参数，可能的输出如下：

```
18:03:59 up  2:08,  2 users,  load average: 0.17, 0.13, 0.18
```

输出的各项的意义如下。

- 18:03:59：是现在的时间。
- 2:08：系统开机到现在已经有 2 小时 8 分钟了。
- 2 users：有两位用户登录到本机（就算只有一个普通用户登录，还要加上一个 root 用户，就是两个用户）。
- load average: 0.17, 0.13, 0.18：最近一分钟，五分钟和十五分钟的系统负载分别是 0.17，0，13，0.18。

5．日历 cal

cal 用来显示特定时间的日历，其使用方式如下：

```
cal [-mjy] [month [year]]
```

该命令的功能是显示日历。若只有一个参数，则代表年份（1-9999），显示该年的年历。年份必须全部写出，如“cal 89”将不会是显示 1989 年的年历。使用两个参数，则表示月份及年份。若没有参数则显示这个月的月历。下面是一些实际使用的例子，可以从这些例子中体会参数的用法。

```
earthfans@10:55 ~$ cal                  #显示本月月历
     四月 2009
一 二 三 四 五 六 日
      1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
earthfans@10:55 ~$ cal -j               #使用-j 参数显示每一天是一年中的第几天
        四月 2009
 一   二   三   四   五   六   日
        91  92  93  94  95              #91 说明四月的第 1 天是今年的第 91 天
 96  97  98  99 100 101 102
103 104 105 106 107 108 109
110 111 112 113 114 115 116
117 118 119 120
earthfans@10:56 ~$ cal -y               #使用 y 参数显示今年年历
                      2009
       一月                二月                三月
一 二 三 四 五 六 日  一 二 三 四 五 六 日  一 二 三 四 五 六 日
        1  2  3  4                   1                   1
 5  6  7  8  9 10 11   2  3  4  5  6  7  8   2  3  4  5  6  7  8
12 13 14 15 16 17 18   9 10 11 12 13 14 15   9 10 11 12 13 14 15
19 20 21 22 23 24 25  16 17 18 19 20 21 22  16 17 18 19 20 21 22
```

```
26 27 28 29 30 31     23 24 25 26 27 28     23 24 25 26 27 28 29
                                             30 31
.............................
       十月                十一月                十二月
一 二 三 四 五 六 日  一 二 三 四 五 六 日  一 二 三 四 五 六 日
        1  2  3  4                   1      1  2  3  4  5  6
 5  6  7  8  9 10 11   2  3  4  5  6  7  8   7  8  9 10 11 12 13
12 13 14 15 16 17 18   9 10 11 12 13 14 15  14 15 16 17 18 19 20
19 20 21 22 23 24 25  16 17 18 19 20 21 22  21 22 23 24 25 26 27
26 27 28 29 30 31     23 24 25 26 27 28 29  28 29 30 31
                      30
earthfans@10:56 ～$ cal 2 2010      #显示 2010 年 2 月月历
     二月 2010
一 二 三 四 五 六 日
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
```

6．多行命令处理程序 xargs

xargs 可以处理由标准输入中传入的命令列表，标准输入可以是 Shell 中终端的输入，也可以是从一个文件中读入。使用格式如下所示：

```
xargs 命令 命令参数
```

但在使用时要用管道或重定向符向 xargs 传入操作的内容。如 ls | xargs vi 或 xargs vi < file。

示例 1：xargs 用在这样的场合：有时候，一个命令输出的结果有很多，如用 ls 列出的所有文件名，但却想用同一种方式处理，如用 vi 进行编辑，这时直接用管道是行不通的。比如“ls |vim”，这样只会得到一个错误信息，而并没有想象中的 vi 打开文件的情形出现。这种情况，就是 xargs 发挥其力量的时候。运行这个命令：

```
ls | xargs vim
```

就会将 ls 列出的文件用 vim 全部打开（vim 有一次打开多个文件的功能）。

示例 2：寻找/tmp 目录中以 core 为名字的所有普通文件，然后用 rm -f 命令删除，使用如下命令：

```
find /tmp -name core -type f -print | xargs /bin/rm -f
```

示例 3：寻找/tmp 目录中以 core 为名字的所有普通文件，然后用 rm -f 命令删除，使用如下命令：

```
find /tmp -name core -type f -print0 | xargs -0 /bin/rm -f
```

这个命令和前一个基本一样，不过 xargs 加了一个参数 O，这个参数的意思是在文件名中有空格和新行的情况下，xargs 也能正确处理这些文件名，而前一个未加 O 参数的就不能正确处理这些情况。

示例 4：使用 xargs 将分行显示的内容显示在一行中，如果不使用 xargs 命令的话，这些内

容会以每项一行的方式显示出来。命令如下所示：

```
cut -d: -f1 < /etc/passwd | sort | xargs echo
```

显示的内容类似于这样：

```
avahi avahi-autoipd backup bin daemon dhcp games gdm gnats haldaemon hplip
irc klog libuuid list lp mail man messagebus news nobody polkituser proxy pulse
root sync sys syslog uucp westnorth www-data
```

示例 5：如果对 xargs 后面接的命令不怎么有信心的话，可以使用 p 参数，让 xargs 执行后面的命令时询问用户是否继续。如下所示：

```
ls | xargs -p vi
```

这个命令输出 ls 的内容后，会停下来，等待用户输入，如果输入 y 或 Y，则执行后面的命令，否则不执行 vi，直接退出。

7．计算器 bc

如果 Shell 中没有一个像样的计算器那就有点不可想象了。bc 就是 Shell 中强大的计算器。其实把 bc 说成是计算器可能有点低估其能力，因为 bc 不只可以充当计算器完成计算功能，还可以使用变量、函数，以及运行预先编制好的语句，bc 的语法类似于 c，熟悉 c 语言的朋友应该很容易上手。下面来看一个具体的运算过程：

（1）启动计算器程序，输入如下命令即可：

```
Bc
#下面是欢迎信息
bc 1.06.94
Copyright 1991-1994,1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
```

（2）在 bc 的命令提示符中输入如下命令，设置精度为 10，意思是运算结果保留 10 位小数：

```
scale=10
```

（3）输入第一个表达式：

```
3+2                     #输入表达式
5                       #输出结果
5/7                     #输入表达式
.7142857142             #可以看到，结果保留了 10 位小数
5^2                     #5 的平方
25                      #结果是 25
5^5                     #5 的 5 次方
3125
```

（4）定义变量，如下所示：

```
a=3                     #定义变量 a 为 3
b=5                     #定义变量 a 为 5
```

（5）比较上步定义的变量 a 与 b 的大小，如果符合 a＞b，则结果 p 为 1，否则为 0。下面是 bc 的输出：

```
p=a>b                          #将 a 与 b 的比较值赋予 p
0                              #输出 p 值，p 是 0
#系统支持许多的比较操作如 ＞、＜、==、!=等
```

（6）测试 a＞b 与 12＜20，如果同时满足这两个条件，则输出为 1。逻辑操作符也有许多，如 a＆＆b 表示须同时满足 a 与 b，a||b 表示满足 a 或 b。

```
p=a>b  && 12＜20               #测试 a>b 的同时是否 12<20
0                              #输出显示没有同时满足上面的条件
```

（7）bc 中可以定义函数，使用 define 命令，如下所示：

```
define d (n) { return (2*n); }
```

（8）使用函数，如下所示：

```
d(6)                           #使用函数
12                             #函数使用结果*/
```

（9）退出的命令是 quit，如下所示：

```
quit
```

bc 还支持许多函数，如下所示。

- s(x)：表示 x 的正弦值，相当于数学中的 sin(x)。
- c(x)：表示 x 的余弦值，相当于数学中的 cos(x)。
- a(x)：表示 x 的反正切值，相当于数学中的 atan(x)。
- l(x)：表示 x 的自然对数值，相当于数学中的 ln(x)。
- e(x)：表示 e 的 x 次幂，相当于数学中的 e^x。

也可以在 Shell 中调用 bc 来进行相关的数学处理，如：

```
pi=$(echo "scale=10; 4*a(1)" | bc -l)
#定义了一个名为 pi 的函数，功能是求π的值，精度是 10 位。然后就可以用“echo $pi”来查看该函数的结果。
```

还可以使用 bc 来编制程序，打开一个文本编辑器，输入如下内容：

```
define f (x) {
if (x <= 1) return (1);
return (f(x-1) * x);
}
```

将该文件保存为 fib，然后退出。再使用命令“bc fib”启动 bc，则加载了该函数。在 bc 中就可以该函数了，如：

```
f(7)                    #调用 fib 文件中的 f 函数
5040                    #输出 7 的阶乘结果是 5040。
```

6.7　强大的手册 man

man 命令无疑是 Linux 或其他的类 UNIX 系统的一个重要特色，其功能主要是显示相关程序的使用手册。得到的内容包括程序的参数列表、各参数的意义，还有程序的功能描述，以及少量的例子。使用也很简单，直接“man 程序”就可以了（其实，man 有许多参数，后面再详细介绍），但其在 Linux 系统中有着极其重要的作用。在不会使用程序的时候，就 man 一下，在 Linux 世界，已经成为一个术语，意思就是用 man 查一下该程序的手册。如“man 一下 ls”，就是在 Shell 中执行“man ls“指令。

man 提供的信息权威、准确，而且文笔优美，对于中国人来说，还是一个学习英语的好地方。man 很重要，但对其的汉化工作一直进展缓慢，所以许多时候不得不阅读英语 man。man 这个名字，来源于“manual”（手册）的前三个字母。man 可以显示的命令帮助信息有 9 部分，其中部分编号的意义如下所示：

- 1：Shell 命令，这也是 man 最主要的部分。
- 2：系统调用，编程时需要查看。
- 3：库函数。
- 4：特殊文件（/dev）。

一般情况下，只有第 1 部分有内容。也就是只能这样使用“man 1 ls”，而不能“man 8 ls”。后者会得到一个错误，说 ls 的第 8 部分没有内容。系统默认的编号是 1。

man 本身也是一个复杂的命令，有许多的参数可供使用，下面就简单地介绍一下。

- man -f man：显示所有 man 文档的编号。
- man -k keywords：在 man 的名称列表和描述内容中查询特定字符，功能同 apropos。
- --a：显示所有符合命令名的手册，默认只显示符合条件的第一个手册。
- -c：重新排版 man 的内容。
- -w 或--path：显示 man 文件所在的目录。

这些参数的使用应该比较简单，就不举例子来说明了。而且 man 命令有一个竞争对手，就是“info coreutils "程序 invocation"”。这个命令的功能也是显示程序的手册，而且一般情况下提供的信息更为详细。但该命令有点复杂，很难记忆，所以使用得很少。

6.8　本章命令列表

本章是本篇最为重要的一章，列举了许多相关的 Linux 基础命令。熟练掌握这些命令，是使用 Linux 必不可少的技能。虽然这对于 Linux 来说，只是其浩如烟海的命令中的一小部分。但对于记忆能力有限的我们而言，已经有点记不住了，下面将本章介绍过的命令按目录层次列于表 6.10，以便查阅。

表 6.10　Shell 基本命令

命　令	小 节 号	功　能	备　注
alias 目的命令＝源命令	6.2.2 使用别名	使用别名来减少输入	可将其写入配置文件

history	6.2.3 历史命令	使用在 Shell 中使用过的命令提高效率	

续表

命　令	小 节 号	功　能	备　注
set -o	6.2.4 键绑定	显示 Shell 可用选项	
bind -p	6.2.4 键绑定	查询已绑定的键	
<，<<，>>，>，>&，\|	6.3 重定向及管道	输入输出重定向及管道	
cat	6.4.1 输出文件内容	显示文本文件内容	
more	6.4.1 输出文件内容	显示文本文件内容，可翻页	
less	6.4.1 输出文件内容	显示文本文件内容，可向上翻页	
head	6.4.1 输出文件内容	显示头几行文本	
tail	6.4.1 输出文件内容	显示最后几行文本	
pg	6.4.1 输出文件内容	分页显示文本	
od	6.4.1 输出文件内容	显示每个字节的内容	
diff	6.4.2 文本文件比较	显示文件间的不同之处	
cmp	6.4.2 文本文件比较	比较文件间的不同	
sort	6.4.3 文本处理	对文本进行排序	
nl	6.4.3 文本处理	给文件添加行号，然后输出	
tr	6.4.3 文本处理	字符转换	
awk	6.4.4awk 文本处理程序	流编辑程序及语言	awk 也是一种程序设计语言
sed	6.4.5sed 流编辑器	流编辑器	
grep	6.4.6grep 文本处理程序	文本内容查找	
file	6.4.7 显示可执行文件详情	显示文件类型信息	
objdump	6.4.7 显示可执行文件详情	显示程序的信息及反编译	
strace	6.4.7 显示可执行文件详情	显示程序的系统调用信息	
ls	6.5.1 显示文件信息命令 ls	列出文件及目录名	
mv	6.5.2 文件移动、复制、删除命令	移动文件	同时也可进行文件或目录的重命名工作
ln	6.5.2 文件移动、复制、删除命令	建立文件链接	可建立符号链接和硬链接
cp	6.5.2 文件移动、复制、删除命令	复制文件	
rm	6.5.2 文件移动、复制、删除命令	删除文件	可将文件按行数或按大小进行分割
split	6.5.3 创建文件链接、分割文件	分割文件	
touch	6.5.3 创建文件链接、分	修改文件信息	

	割文件		

续表

命　令	小 节 号	功　能	备　注
ln	6.5.3 创建文件链接、分割文件	创建文件链接	
pwd	6.5.4 目录操作	显示当前目录	
cd	6.5.4 目录操作	改变当前目录	
mkdir	6.5.4 目录操作	创建目录	
rmdir	6.5.4 目录操作	删除空目录	
find	6.5.3 文件查找命令 find	寻找文件	
which	6.5.3 快速文件查找命令及特殊查找命令	寻找程序文件	
whereis	6.5.3 快速文件查找命令及特殊查找命令	寻找文件的三部分	可寻找文件的程序文件、手册、源代码
locate	6.5.3 快速文件查找命令及特殊查找命令	快速查找文件	
sleep	6.6 其他命令	程序休眠一段时间	
time	6.6 其他命令	显示程序的系统开销	可显示程序运行时间，内存占用等信息
date	6.6 其他命令	显示更改当前日期	
update	6.6 其他命令	获得系统相关时间信息	
bc	6.6 其他命令	功能强大的计算器	同时也是一个编程环境
cal	6.6 其他命令	显示日历	与之类似的一个命令是 ncal
xargs	6.6 其他命令	处理多行命令	可以通过管道将多个参数传递给命令
man	6.7 强大的手册 man	强大的在线手册	

6.9　小结

本章是本书的重点，详细介绍了 Linux 最有特色也是最为重要的 Shell 命令行程序，共介绍了约 50 条最为实用、常用的命令行程序。通过这些程序，可以完成绝大部分 Linux 系统的操作。如果熟练掌握了这些命令，便可以从很大程度上提高工作效率。这些命令可能在开始使用时需要花费一定的精力去记忆，但实际使用的效果绝对会让记忆命令的时间不至于浪费。

第 7 章　文 件 管 理

文件管理的主要目的就是有效地操作文件。文件管理的内容比较丰富，包括文件的建立、删除等。Linux 提供了两种方式进行文件管理，一是命令行方式，二是图形界面软件的管理方式。本章主要介绍了图形界面软件管理的方法。

7.1　Linux 的目录

对于熟悉 Windows 目录结构的用户来说，Linux 的目录结构显得有些怪异。但其实熟悉该用法后，会感觉到这样的目录结构有其优越性。本节主要介绍 Linux 的目录结构，同时对一些重要的目录也进行了介绍。

7.1.1　Linux 目录结构

Linux 采用了树形目录来管理所有的目录。而且在该树形目录中，只有唯一的一个根目录“/”，其余目录均是根目录的子目录，如图 7.1 所示。Linux 将所有的硬件设备也作为文件来管理，如所有的磁盘分区均是“/dev”目录中的一个文件，该目录中以 sda 或 hda 开头的文件即为硬盘的分区。在笔者的系统中，使用如下命令。

```
ls /dev/sd*
```

得到的输出如下所示。

```
/dev/sda   /dev/sda2  /dev/sda6  /dev/sda8
/dev/sda1  /dev/sda5  /dev/sda7  /dev/sda9
```

这些以 sda 开头的文件就代表了系统中硬盘的各个分区。以上的输出内容说明系统中安装有一块硬盘（sda），该硬盘有两个主分区（sda1-2），5 个逻辑分区（sda5-9）。

在“/”根目录中，有许多子目录，这些子目录各有其不同的用途，其包含的内容简介如下。

- /usr：包含所有命令、类库、系统帮助手册，即包含了系统运行中不会修改的内容。
- /var：包含系统运行中会修改的内容，如打印队列、邮件列表等，还有系统中软件的 log 信息。
- /home：用户的主目录所在地，实际用户的主目录是该目录中的次级目录，如“/home/zhangsan”。对于每个用户而言，均有专属于自己的一个用户目录。

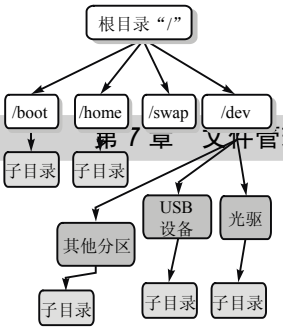


图 7.1 Linux 目录结构示意图

- /dev：设备文件，如硬盘存储器、光驱、USB 设备等。设备文件用特定的约定命名。
- /tmp：临时文件。该目录中的文件会在每次启动后被清空。
- /boot：系统内核所在地，如果安装了启动管理器的话，其也会安装在这个目录下。
- /mnt：临时加载文件系统的安装点。
- /proc：其他文件系统的安装点。
- /etc：目录包含很多文件。许多网络配置文件也在/etc 中。
- /bin：引导启动所需的命令及用户在使用过程中可能会用到的程序。
- /sbin：与/bin 类似，但其一般是提供给 root 用户使用。
- /etc：大多数程序的配置文件。
- /root：root 用户的主目录。
- /lib：根文件系统中程序所需的共享库。
- /proc：由内核建立，只存在于内存中，而且只在需要时生成相关文件，用于提供关于系统的信息。

7.1.2 Linux 的一些重要目录简介

7.1.1 节的简介对于其复杂的功能来说，是过于简单了，下面将挑选一些目录，将其功能列举出来，以使读者对 Linux 结构有更深入的了解。

1．/etc 目录

/etc 目录包含了大部分的系统配置文件，一般情况下，需要对系统设置的话，均可以到/etc 目录中进入相应的目录寻找配置文件。该目录中部分目录及其作用如下所示。

- /etc/rc、/etc/rc.d、/etc/rc*.d：系统启动时读取自动运行程序与设置启动级别。
- /etc/passwd：用户的密码及其相关信息。
- /etc/fstab：启动时自动加载的文件系统。
- /etc/group：用户组的相关信息。
- /etc/issue：登录提示信息。
- /etc/mtab：当前挂载的文件系统列表。
- /etc/profile：设置终端启动时运行的程序。
- /etc/timezone：时区。
- /etc/environment：环境变量。
- /etc/updatedb.conf：文件检索数据库配置信息。
- /etc/iftab：网卡 MAC 地址绑定。
- /etc/hostname：主机名。
- /etc/resolv.conf：域名解析服务器地址。
- /etc/network/interfaces：网卡配置文件。

2．/usr 目录

/user 目录包含了用户相关的内容，如用户的应用程序、用户需要查看的相关文档等。一般情况下需要查找用户使用的程序或文档可查看该目录的相关内容。该目录部分目录及说明如下

所示。

- /usr/X11R6：X Window 系统的所有文件。
- /usr/bin：大多数的命令，还有些命令安装在/bin 或/usr/local/bin 中。
- /usr/sbin：供用户使用的系统管理命令。
- /usr/man：用 man 命令列出的手册存放地。
- /usr/info：GNU 信息文档。
- /usr/doc：其他文档。
- /usr/include：C/C++编程所需的头文件。
- /usr/local：用户安装的软件所在地。
- /usr/src：程序源代码。

3．用户主目录的一些配置文件

Linux 每一个用户均有其专用的主目录，该用户的配置文件、启动设置等均在该目录。用户主目录用“～”来指代。如果用户是 zhang3，则其主目录是“/home/zhang3”。该目录中一些常见的配置文件及其说明如下所示。

- ～/.bash_logout：在退出 Shell 时执行的代码。
- ～/.bash_history：Shell 的命令历史列表。
- ～/.bashrc：Bash 启动执行的代码。
- ～/.emacs：Emacs 启动时执行。
- ～/.profile：如果～/.bash_profile 和～/.bash_login 不存在，Bash 会将～/.profile 作为～/.bash_profile 看待。
- ～/.vimrc：vim 的配置文件，和.exrc 功能一样。
- ～/.local/share/Trash：Ubuntu 回收站的目录。其中有两个目录，一个名为 file，存放的是实际在回收站中的文件；一个名为 info，是对回收站中文件描述的信息。

7.2 文件系统格式

文件系统是计算机操作系统中一个很重要的部分。文件系统在底层实现了对于文件的组织、管理，并且通过人机接口向用户提供了操作文件的方法。本节将简单介绍文件系统的概念，以及 Linux 中比较常用的几种文件系统格式。

7.2.1 文件系统概述

对于用户而言，如果硬盘上充满了数据，但没有向用户提供一种适当的方式或接口来访问这些数据，那么这些数据是毫无意义的。操作系统中文件系统就负责管理这些硬盘（以及其他任何存储设备）上的数据信息，如对存储器空间进行组织和分配、管理文件的存储等工作。用户使用的任何功能（文件管理、执行程序等）都是建立在文件系统的基础之上的，而且不同的操作系统对于文件系统有着不同的实现，如图 7.2 所示。

由图 7.2 可以看出，对于通用格式的文件，如音、视频、纯文本等格式的文件 Linux 与 Windows 操作系统均可正常打开使用，而 Linux 私有的格式文件，如类库、Linux 平台应用程序等则只能在 Linux 文件系统中打开使用，反之亦然。

磁盘在作为文件系统使用前，应该被初始化，以记录文件系统的数据结构。建立数据结构后，如何访问该磁盘、如何管理文件都已被设定为一种特定的方式，这个过程就叫建立文件系统。UNIX 及类 UNIX 操作系统的文件系统虽然有许多种，但这些文件系统均具有类似的通用结构。文件系统的数据结构主要包括超级块（superblock）、i 节点（inode）、数据块（data block）、目录块（directory block）等。下面分别予以简单的介绍。

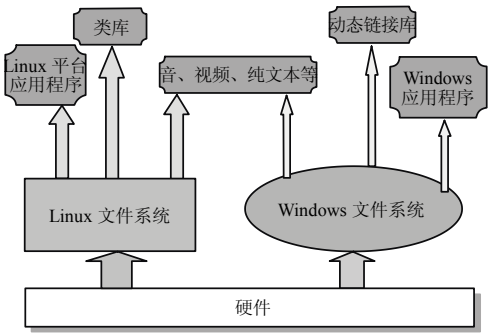


图 7.2　不同的操作系统有不同的文件系统

1．超级块

超级块包含了分区上的文件系统的整体信息，是文件系统的控制块。超级块包括文件系统的状态、尺寸和空闲磁盘块等信息。具体内容包括文件系统大小、可用块数目、可用块索引表、i 节点数目、i 节点索引表等。超级块是系统管理文件存储空间、回收存储空间的依据。如果超级块被损坏，则会造成系统无法识别该文件系统。每个分区包含一个超级块。

2．i 节点

i 节点区存放 i 节点，i 节点是文件管理的主要结构。源代码中对于 i 节点结构定义如下：

```
struct dinode
{
 ushort di_mode;     /*文件类型+用户权限*/
 short di_nlink;     /*文件链接数*/
 ushort di_uid;      /*属主用户 id*/
 ushort di_gid;      /*属主用户组 id*/
 off_t di_size;      /*文件大小*/
 char di_addr[40];   /*文件数据区起始地址*/
 time_t di_atime;    /*最后访问时间*/
 time_t di_mtime;    /*最后修改时间*/
 time_t di_ctime;    /*创建时间*/
};
```

i 节点内包含了文件的权限、类型，以及文件的时间信息等，但是却没有包含文件名，也没有包含实际文件的内容，而是将其指向数据块。可以使用“ls -i”查看文件的 i 节点的信息

3．目录块

目录块中包括文件名和文件的 i 节点编号。

4．数据块

数据块中包含了文件实际的数据。

5．文件系统的状态

文件系统有两种状态，clear、dirty。clear 代表文件是没有做过任何修改，如果文件被修改，则其状态会由 clear 变成 dirty。而将更改提交后，dirty 状态会重新回到 clear 状态。

6．文件系统的格式

Linux 的文件系统一般采用 ext2、ext3 或 ReiserFS，各文件系统均使用了同样的数据结构，这就使得一种统一的访问方式成为可能。虚拟文件系统（VFS）实现了此项功能。对于用户而言，可以不用理会文件系统实现的细节。即不管文件系统是 ext2、ext3 或 ReiserFS，用户访问方式是一样的。

- ext2 是标准的 Linux 文件系统格式，但最近有被 ext3 替代的趋势。ext2 不是日志文件系统，所以如果系统硬件发生故障，重新启动时需运行 fsck 工具。对于大型系统而言，这可能需要花费很长的时间。
- ext3 是 ext2 的升级版，提供了日志文件功能。就是在系统操作时，记录了对于元数据的修改，如果系统崩溃，可以通过记录的日志文件快速修复系统。
- ReiserFS 支持可变大小的文件创建，更为灵活些，同样也是日志文件系统。

7．日志文件系统

使用日志文件，其好处是系统崩溃后恢复速度快，但缺点是系统运行过程中，记录日志文件会加大系统开销，这样就会从某种程序上降低系统性能。ext3 提供了控制日志文件的功能，可以通过设置使得日志中只记录一些特定的信息。不过，总体而言，ext2、ext3、ReiserFS 在各种场合性能基本相当。本节接下来的部分将具体介绍文件系统的知识。

7.2.2 ext2、ext3 文件系统

到目前为止，Linux 各发行版一般默认的文件系统是 ext3。虽然随着技术的前进，这种状况也许会发生改变，但既然 ext3 得到了如此广泛的应用，还是有必要了解其特性的。

1．ext2 文件系统

ext2 英文名是 second extend file system，意思是第二个扩展文件系统。它最初设计的目的是为了代替 ext 文件系统。

早期的 Linux 使用 Minix 文件系统，文件大小的上限是 64 兆字节，文件名长度也限制在 14 个字符。1992 年 4 月发布了第一个使用虚拟文件系统的 Linux。虚拟文件系统层被添加到 Linux 内核，它提供了一种统一的文件访问层，并解决了最大分区大小和文件名长度的限制。但是，它仍存在不少问题。所以，1993 年 1 月开始开发第二个扩展文件系统（ext2）。

由于 ext2 具有很灵活的的扩展性，所以许多新扩展的虚拟文件系统的 API 均在其中得到实验。如 POSIX 的 ACL 等都是首先在 ext2 文件系统实施的。在 Linux 内核 2.6 之前，ext2 文件系统最大的文件大小已经扩大到 2TB。ext2 文件系统的空间被分成块，然后将块组织成块组，这样做是为了减少磁盘碎片和减少读取大量文件时的寻道时间。

至今，ext2 仍是可引导的 USB 闪存设备和其他固态硬盘的首选格式。因为 ext2 不用向日志里写数据，减少了对于闪存芯片的读取，增加了这些设备的寿命。ext2 的一些主要性能指标如下所示。

- 分区类型号：0x83。
- 最大文件大小：2-64TB。

- 最大文件数量：1018。
- 最大文件名长度：255 个字符。
- 最大卷大小：16-32TB。
- 文件名中不可使用“/”字符。
- 支持的日期范围：1901 年 12 月 14 日～2 038 年 1 月 18 日。
- 支持的操作系统：Linux、BSD、Windows、Mac OS X。
- ext2 不是日志文件系统。

2．ext3 文件系统

ext3 是一个日志文件系统，Linux 内核从 2.4.15 版本开始将其合并到内核功能中。它是 ext2 的升级版本，可以将系统中文件的更改记录在日志文件中。从某种意义上说，ext3 是有日志功能的 ext2。因为其内核及功能是如此的相似，使得两种系统之间的转换是非常容易的。由于 ext2 在 Linux 世界中极其流行，所以这个特点也给 ext3 扩大市场份额带来了方便，而且其 CPU 的使用要低于 ReiserFS 等文件系统。

7.2.3　日志文件

日志文件系统是现代 Linux 操作系统的一个重要安全特性。但为了满足不同的需要，一般 Linux 文件系统均提供了三种权限，如下所示三种权限的安全性依次降低，但速度依次提高。用户可以根据安全及速度需求酌情选择。这三种权限相应的模式是：

- journal 日记模式：将数据块和元数据的写入均记入日志中，是最安全但也是最慢的模式。因为所有的数据都要写两次，一次写到目标数据区，另一次写到日志中。
- orderd 顺序模式：这是大多数发行版的默认模式。这种模式中，只记录元数据的修改情况，但在写元数据之前会先将数据块的内容写入磁盘中。
- writeback 回写模式：这种模式是最快的模式。该模式只简单地记录元数据，而不管是否已将数据写入数据块中。

ext3 文件系统的一些相关的系统操作如下所示。

1．将 ext2 文件系统转换为 ext3 文件系统

如果将 ext2 文件系统转换为 ext3 文件系统，可以使用如下命令：

```
tune2fs -j /dev/sda6
```

这样就可以将原来的 ext2 文件系统转换为 ext3 文件系统。由于 ext2 与 ext3 之间的高度兼容性，这个操作在转换前甚至不必卸载/dev/sda6。转换完后，要记得将系统中所有使用该文件系统格式的地方做一些修改，如/etc/fstab 中关于该分区格式的描述。

2．指定 ext3 日志的存放位置

如果要创建一个 ext3 文件系统，并将日志文件存放于指定的位置，可以运行如下命令：

```
mke2fs -J device=/dev/hda7 /dev/hdb5
```

就在 had7 分区创建了一个 ext3 文件系统，并指定将日志文件存入 hdb5 中。

3．ext3 文件系统的修复

在 ext3 文件系统出现故障、文件系统被破坏时，可以使用 e2fsck 来尝试进行修复。不过

在修复 ext3 文件系统前，需要先将其卸载。这个操作很具危险性，不建议普通用户使用，步骤如下：

```
umount /dev/hda7
e2fsck -fy /dev/hda7
```

7.3 加载其他文件系统

在 Linux 中，其他格式的分区、外接的硬件设备（如光驱、USB 设备）均被作为文件被加载到系统中使用。本节介绍了如何对这些设备进行加载的方法，以及如何在 Linux 中实现启动时自动加载其他分区的方法。

7.3.1 加载与卸载文件系统

为了使用其他文件系统或其他硬件设备，需要先将其加载到 Linux 文件系统。将其加载后，就可像使用其他普通文件一样使用这些设备了。在 Linux 中，加载设备主要使用 mount 命令来实现，mount 会将设备挂载到一个已经存在的目录中，所以在使用 mount 加载设备前，应该先创建该目录。挂载设备后，该目录原有的内容会无法访问，但内容并未消失，在取消挂载后，就又可重新访问了。

mount 命令用来加裁一个文件系统，一般用两步即可完成加裁。先建立用来加载的目录，然后使用 mount 进行加裁，具体步骤如下。

（1）建立一个目录用来作挂载点，使用命令“mkdir -p /mnt/win2000”。

（2）将第一块硬盘的第一个分区挂载至该挂载点“mount -t vfat /dev/sda1 /mnt/win2000”。

mount 命令与其他命令一样，有许多参数，其格式如下：

```
mount [-t 文件类型] [-O 选项] 设备 本地目录
```

参数-t 指定文件系统的类型。如果不指定，mount 会自动选择一个类型，但有时自动选择功能会出现错误，所以最好自己指定，常用文件类型如下所示。

- iso9660：光盘或光盘镜像。
- msdos：fat16 文件系统，早期的 Windows 及 DOS 中常用的文件系统。
- vfat：fat32 文件系统，Windows9x 系统使用的文件系统。
- ntfs：ntfs 文件系统，Windows 2000 及 Windows XP 等使用的文件系统。
- smbfs：Windows 文件网络共享，挂载 SMB 时使用。
- nfs：类 UNIX 系统文件网络共享，挂载 NFS 时使用。
- 参数“-o”用来描述设备的挂载方式，常用的选项如下所示。
- loop：把设备当成硬盘分区挂载，常用来挂载光盘、ISO 文件。
- ro：用只读方式挂载设备。
- rw：用读写方式挂载设备。
- iocharset：指定挂载的文件系统所使用的字符集，用来在英文系统中挂载中文分区。

参数“设备”指定要挂载的设备。如上例中，设备是“/dev/sda1”。

参数“本地目录”指定要将设备挂载到什么目录。如上例中，本地目录就是“/mnt/win2000”。

7.3.2　挂载光盘镜像

使用 Linux 时，会经常遇到需要处理镜像文件的情形。一方面从网上下载的操作系统安装文件（如 Ubuntu 的系统安装文件）是光盘镜像格式，另一方面由于近年来磁盘技术的巨大进步，新的计算机系统都配备了大容量的磁盘系统。在 Windows 下，许多人都习惯把软件和资料做成光盘镜像文件通过虚拟光驱来使用。这样做有许多好处：一是减轻了光驱的磨损；二是现在硬盘容量巨大存放几十个光盘镜像文件不成问题，随用随调十分方便；三是硬盘的读取速度要远远高于光盘的读取速度，CPU 占用率大大降低。其实 Linux 系统下制作和使用光盘镜像比 Windows 系统更方便，不必借用任何第三方软件包。

1．挂载光盘镜像

挂载镜像文件的方法很简单，只需如下两步。

（1）创建镜像文件挂载点（也就是挂载目录）：

```
mkdir /mnt/source
```

（2）将刚才创建的 ISO 文件挂载到创建的挂载点，就可以像打开普通的目录一样打开 /mnt/source 了：

```
mount -o loop -t iso9660 ～/source.iso /mnt/source
```

2．制作镜像文件

使用系统命令，可以很容易将光盘甚至是系统中某些目录制作成 ISO 文件。从光盘制作 ISO 文件很简单，将光盘放入光驱后，执行下面任意一条命令，即可将光盘中的内容制作为一个在当前用户主目录中的 source.iso 文件。

- cp /dev/cdrom ～/source.iso：直接使用 cp 命令就可将目录、文件复制到一个 iso 文件中，这样就建立了一个 iso 镜像。
- dd if=/dev/cdrom of=～/source.iso：也可用 dd 命令来制作一个 iso 镜像。
- mkisofs -r -J -V article -o ～/source.iso ～/Document：使用 mkisofs 也可完成创建 iso 文件的创建。本例将～/Document 目录下所有的目录和文件制作成光盘镜像文件～/source.iso，光盘卷标为 article。

7.3.3　挂载移动硬盘、U 盘

一般来说，系统只要正确加载了 USB 的相关识别模块。在移动硬盘、U 盘插入系统后，会被自动识别。现在的发行版一般都可正确识别各种 USB 设备，并将其作为 SCSI 设备来对待。USB 设备挂载一般需要分三个步骤，先查看 USB 设备号，再创建一个用来挂载的目录，最后挂载即可。具体步骤如下。

（1）用“sudo fdisk -l”或“cat /proc/partitions”查看系统的存储设备列表。笔者的系统输出如下（以“#”开头的文字是笔者所加）：

```
Disk /dev/sda: 100.0 GB, 100030242816 bytes
255 heads, 63 sectors/track, 12161 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0xf0b1ebb0
```

```
   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1        1825    14659281    7  HPFS/NTFS
/dev/sda2            1826       12161    83023920    f  W95 Ext'd (LBA)
/dev/sda5            1826        3100    10241406    7  HPFS/NTFS
/dev/sda6            3101        3116      128488+  83  Linux
/dev/sda7            3117        4332     9767488+  83  Linux
/dev/sda8            4333        4394      497983+  82  Linux swap / Solaris
/dev/sda9            4395       12161    62388396   83  Linux
#上面列出的是 sda（系统第一块硬盘）的分区情况
Disk /dev/sdb: 64 MB, 64225280 bytes
#检测出来的第二块硬件设备，大小为 64MB，这就是要挂载的 USB 设备。这里的内容，根据系统不同，会有不同的输出
8 heads, 32 sectors/track, 490 cylinders
Units = cylinders of 256 * 512 = 131072 bytes
Disk identifier: 0×00000000
   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1         490       62700+   6  FAT16
#/dev/sdb1 就是实际要挂载的分区号
```

（2）创建挂载点，使用如下命令即可：

```
mkdir -p /mnt/USB
```

（3）挂载 USB 设备，使用如下命令即可：

```
mount -t vfat /dev/sdb1 /mnt/USB
```



如果中文出现乱码，则用 iocharset 指定的编码挂载 USB 设备，命令是“mount -t vfat -o iocharset=cp936 /dev/sdb1 /mnt/USB”。

7.3.4 挂载 SMB 文件系统

Linux 与 Windows 系统共享文件与目录的一个常见方式是使用 SMB 文件系统。在 SMB 服务安装设置成功后，就可以在 Linux 中将 Windows 共享目录挂载在 Linux 中正常使用了。其步骤如下。

（1）先创建挂载点：

```
mkdir -p /mnt/winxp
```

（2）获得 Windows 系统的用户名与密码，如用户名是 zhang，密码是 123456。同时设 Windows 的主机 IP 地址为 192.168.2.23，共享目录名是 share。就可用如下方式挂载：

```
mount -t smbfs -o username=zhang,password=123456 //192.168.2.23/share$ /mnt/winxp
```

7.3.5 挂载 NFS 文件

在 Linux 与其他类 UNIX 系统共享文件，可使用 NFS 文件系统。主机和客户机的 NFS 服务均配置好后，就可使用下列步骤挂载 NFS 文件。

（1）创建挂载点：

```
mkdir -p /mnt/nfs
```

（2）设 Linux 主机 IP 地址是“192.168.2.2”，其设置为 NFS 共享的目录是/share，并将其访问权限设置为可读可写，则用如下命令：

```
mount -t nfs -o rw 192.168.2.2:/share /mnt/nfs
```

7.3.6　挂载所有文件系统

如果已经将要挂载的文件系统写入了配置文件（位于/etc/fstab）中，那么可使用如下命令将其全部挂载：

```
sudo mount -a
```

该命令的功能是挂载所有在 fstab 表中定义的文件系统。

7.3.7　卸载文件系统

在很多时候，要对挂载的文件系统进行卸载，如需要删除光盘镜像文件、要拔出 USB 设备、或者光驱停止响应无法取出光盘等。在这时候，就可以使用 umount 命令卸载文件系统。umount 的使用方法一般类似于“umount /mnt/USB”，在该命令中“/mnt/USB”是 mount 命令的挂载点（见本节第 3 项“挂载移动硬盘、U 盘”中相关内容）。也可直接以分区号来卸载文件系统，如“umount /dev/sdb1”。

如果卸载失败，可能是由于其他很多原因，但解决方法很简单，使用 umount 的 lazy 模式：“umount -l /dev/USB”，就可以解决无法卸载的问题。umount 还有几个比较有用的参数是：

- -f：强制卸载。用于服务器名无法解析或者卸载正在使用的文件系统。可用于卸载 NFS 文件，如“umount /mnt/nfs”；
- -n：卸载后不更新“/etc/mtab”表；
- -a：卸载所有已加载的文件系统。

7.3.8　开机自动加载

如果系统中存在几个不同的文件系统，而在使用中又需要经常使用该文件系统中的内容，就可以将其设定为开机就自动挂载。系统中管理自动挂载的配置文件为/etc/fstab。与其他配置文件一样，其内容是纯文本格式的，可用“cat /etc/fstab”查看。一个典型的 fstab 文件如下所示（以“#”开头的为注释）。

```
    # /etc/fstab: static file system information.
    # <file system> <mount point>   <type>  <options>       <dump>  <pass>
    proc            /proc           proc    defaults        0       0
    # /dev/sda7
    UUID=92ac6170-07a4-4723-a27d-44864cde4a8a /         ext3 relatime,errors=
remount-ro  0   1
    # /dev/sda6
    UUID=4b241a16-6c2d-45ad-aea7-e0f9df00c0c6 /boot    ext3 relatime  0   2
```

```
    # /dev/sda9
    UUID=1a6603c2-bdc1-45ca-9461-b8658d7c8918 /home  ext3 relatime  0   2
    # /dev/sda8
    UUID=01ae04e8-3ef0-4a74-93ea-76ea7adb1627 none    swap  sw        0   0
    /dev/scd0       /media/cdrom0                   udf,iso9660  user,noauto,
exec,utf8    0   0
```

第二行说明了各项的意义，在 fstab 中的各项有 6 列。其意义依次如下所示。

1．file system（文件系统）

要挂载的文件系统。其值可以用设备号来表示，也可以直接使用 UUID 来表示。设备号可以使用“sudo fdisk -l”命令来查看系统中所有硬盘的设备号。一个典型的输出如下所示：

```
  Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1        1825    14659281    7  HPFS/NTFS
/dev/sda2            1826       12161    83023920    f  W95 Ext'd (LBA)
/dev/sda5            1826        3100    10241406    7  HPFS/NTFS
/dev/sda6            3101        3116      128488+  83  Linux
/dev/sda7            3117        4332     9767488+  83  Linux
/dev/sda8            4333        4394      497983+  82  Linux swap / Solaris
/dev/sda9            4395       12161    62388396   83  Linux
```

查到设备号就可直接使用，如使用“/dev/sda6”来加载第一块硬盘的第二个逻辑分区。但使用设备号有个问题，就是如果增加了硬盘，则设备号可能会发生改变，会造成错误，需要重新进行设置。而使用 UUID 号，则不存在这个问题。硬盘被分区、格式化后，该分区的 UUID 号就固定了，不会再变。得到硬盘的 UUID 号也比较容易，只需要执行如下命令：

```
    ls -l /dev/disk/by-uuid
    #得到的输出类似于
    lrwxrwxrwx 1 root root 10 2008-11-07 22:50 01ae04e8-3ef0-4a74-93ea-76ea
7adb1627 -> ../../sda8
    lrwxrwxrwx 1 root root 10 2008-11-07 22:50 1a6603c2-bdc1-45ca-9461-b8658
d7c8918 -> ../../sda9
    lrwxrwxrwx 1 root root 10 2008-11-07 22:50 4b241a16-6c2d-45ad-aea7-e0f9
df00c0c6 -> ../../sda6
    lrwxrwxrwx 1 root root 10 2008-11-07 22:50 92ac6170-07a4-4723-a27d-44864
cde4a8a -> ../../sda7
    lrwxrwxrwx 1 root root 10 2008-11-07 22:50 BCD05A69D05A2A42 -> ../../sda5
    lrwxrwxrwx 1 root root 10 2008-11-07 22:50 FA947EC7947E85C3 -> ../../sda1
```

在时间信息之后、设备号之前的字符串就是该设备的 UUID 号，如 sda6 的 UUID 是“4b241a16-6c2d-45ad-aea7-e0f9df00c0c6”。要让 sda6 在启动时加载，fstab 中第一列的内容就可以写为：“UUID=4b241a16-6c2d-45ad-aea7-e0f9df00c0c6”。

2．mount point（挂载点）

表示要将文件系统挂载的位置，该目录必须已经存在。如果挂载点包含空格的话，可以用 40 来表示空格。

3．type（文件类型）

要挂载的文件系统的类型，可以是 7.3.1 节 mount 命令中介绍的任意一种文件系统。如 vfat、ntfs。

4．option（选项）

指定加载该文件系统需要使用的特定参数选项，多个参数间用逗号分隔。大多数情况下使用 defaults 就可以。其他常见的选项如下所示。

- auto：开机自动挂接，这是默认值。
- noauto：开机不自动挂载。
- exec：可以运行文件系统中程序。
- noexec：不能运行文件系统中程序。
- nouser：只有 root 用户可挂载。
- ro：以只读方式挂载。
- rw：以可读、写方式挂载。
- user:：非 root 用户也可以挂载。
- defaults ：包含了这些选项 rw、suid、dev、exec、auto、nouser 和 and async。

5．dump（转储）

设定备份的频率，单位为天。如设置为 2，就是每两天备份一次。

6．pass（扫描）

设置启动时扫描的顺序。只有“/”目录应该设置为 1，其他目录应该设置为 2。如果在启动时不需要扫描，则将其设置为 0。

明白了这些选项的意义后，就可以自己加入内容了。如要使/dev/sda5 在启动时加载到“～/winsoft”目录（该目录必须存在），就可以在 fstab 文件中写入如下任一种内容：

```
#方法 1：使用 UUID
UUID= BCD05A69D05A2A42  /home/用户名/winsoft  ntfs  default  0  0
#方法 2：使用设备号
/dev/sda6  /home/用户名/winsoft  ntfs  default  0  0
```

7.4　文件管理器

文件管理，就是提供操作文件的方法，如打开、编辑、查找文件等。Ubuntu 中默认的图形界面文件管理器是 Nautilus。这款文件管理器提供了强大的文件管理功能，如对于其他文件系统的自动加载、USB 设备的自动加载等，还具有接管桌面设置桌面背景等功能。本节将简单介绍这些功能。

7.4.1　文件管理器界面

对于普通用户而言，对文件的操作可能大部分都是在图形界面下实现的。所以图形界面的文件管理器就显得极其重要了。Ubuntu 提供的这款文件管理器不但功能强大，而且操作方式直观，对于初学者来说，很容易上手。

打开文件管理器的方法很简单，在主菜单 places 中选择任一项，如 home，就在图形界面中打开了用户的主目录。或者在终端中运行 nautilus 也同样可以打开该文件管理器。文件管理器打开后界面如图 7.3 所示。



图 7.3　文件管理器界面

文件管理器的界面看上去和 Windows 的文件管理器很类似，但其有许多功能是很有特色的，如可以显示文本文件和视频文件的预览信息，如图 7.3 所示，系统可以正常播放的视频文件均显示了其视频文件的第一帧的图像。Wall-E_larceny.srt 是纯文本文件，也显示其前几行文本的内容。除了界面上的这些特点外，Nautilus 还有其他功能，分别介绍如下。

1．菜单区

菜单区是文件管理器的菜单，包含了绝大部分的功能。

- File 菜单，可以完成创建文件夹、新建文件、打开文件等功能。
- Edit 菜单，可以完成文件的复制、粘贴等功能，还可以对文件进行加密、添加数字签名等功能。
- Go 菜单：可以快速访问某些特定的目录，如我的文档等。
- Bookmarkers 菜单：可以将经常访问的目录添加到该菜单中以便以后访问。
- Help 菜单：获得系统帮助。

2．工具栏

设原始目录是“～/mybook”，并由地址栏输入地址转到当前目录“/usr/bin”，则工具栏的按钮的作用如下所示。

- Back：返回到上一次访问的目录。本例退回到“～/mybook”。
- Forward：执行完“Back”后如果又想回去，可按此按钮。本例回到“/usr/bin”。
- Up：到当前目录的上级目录。本例转到“/bin”。
- stop：在执行一个耗时较长的操作时，可按此按钮取消操作。
- Reload：刷新当前目录。如果在终端中建立了一个目录，而图形界面中未反映出来，可按此按钮使其刷新文件信息，显示当前目录最新的变动情况。
- Home：进入用户的主目录。如用户为 zhang，则回到“/home/zhang”目录。

- Computer：进入系统硬件管理界面，显示了当前系统所有的文件系统，包括未挂载的，可以双击未挂载的文件系统使其挂载。
- Search：搜索文件。

3．目录

该软件中，目录显示是以按钮形式出现的。这样显示可能会让部分用户感到不适应，但采用这种显示形式有其优点，即可以方便地回到任一级目录。

4．目录树

目录栏只显示了当前的目录项，要打开其他的目录，比较好的方式就是在左侧的目录树中寻找相关的目录。目录树中的内容与菜单 go 中内容基本相同，但访问比较容易。

5．文件显示区

显示了当前目录中的所有目录与文件，并根据设置显示或不显示隐藏文件。如果文件是图片格式、视频格式或是纯文本格式，将会显示其缩略图。

6．放大文件图标

在“目录”右侧，“工具栏”下方有两个放大镜图标，100%。单击左侧带“－”的按钮会将文件图标缩小显示，而右侧的“＋”按钮会将文件图标放大显示。中间的数字显示了当前显示的比例。

7．改变文件显示方式

以图标方式显示文件虽然可以预览其内容，但有个问题就是显示的内容较少。如果要显示更多的项目，并且显示文件详细资料，就可以单击图 7.4 列表方式显示文件中所示地方，将文件管理器显示文件的方式更改为“View as List”，这样就可以直观地查看文件名、文件大小、文件类型、创建时间等信息了。

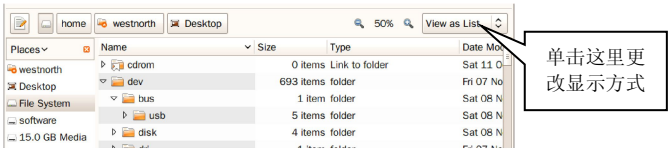


图 7.4　列表方式显示文件

前面介绍了文件管理器的界面，下面介绍文件管理器的基本操作。

7.4.2　打开文件

文件管理器最主要的功能是打开文件。要打开文件之前，首先要定位到目标文件。如果知道文件的路径，如要打开“/etc/apt/sources.list”，可先在目录树中单击“File System”，打开根目录，然后在“文件显示区”双击 etc 目录图标就会打开 etc 目录，再按同样方式打开 apt 目录。依次打开到 apt 层后，界面如图 7.5 所示。



可以在打开目录和文件时输入文件名的前几个字符来快速定位目录和文件。如在图 7.5 的界面中，直接在键盘输入 sou，就会在屏幕右下角出现一个小文本框。而且随着字符的输入，当前文件的焦点会随之在相应的文件上出现。图 7.5 中输入 sou，焦点停留在“sources.list.d”目录，因为其前几个字符也是“sou “，这时只需按键盘上的下方向键，就可将焦点移动到 sources.list 文件上。这种方式在大量文件中筛选时特别有用。

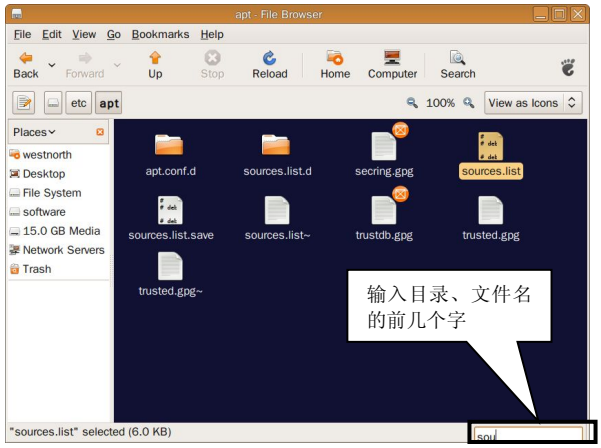


图 7.5　打开文件

要打开的文件找到了，相应的打开文件的方式也有以下几种。

- 选择 File|Open with softsource 命令。注意该菜单项的内容一般是 Open，但随着文件类型的不同会有所不同。如选择视频文件，可能显示的内容就是“Open with MPlayer movie player”。
- 按组合键 Ctrl+O。
- 直接按下 Enter 键。
- 双击该文件。

7.4.3　更改文件的打开方式

一般情况下，打开文件均按系统默认的方式进行操作。如文本文件默认由 gedit 程序打开，html 默认由 Firefox 打开等。有时为了某种需要，可能需要使用另一个程序打开文件。如为了查看 html 的源码，可能就需要使用文本编辑器打开 html 文件；或者要使用 gvim 打开文本文件等。这时，就需要更改系统打开文件的方式。

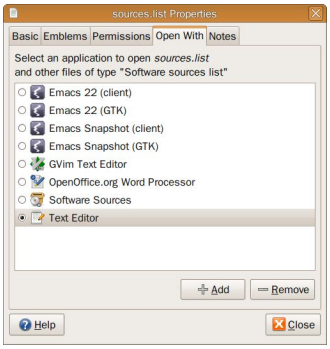


图 7.6　选择默认打开文件的方式

更改文件的打开方式，方法很简单。在要打开的文件上单击右键，选择弹出菜单中的“Open with”菜单项，会再现一个次级菜单，里面列出了可供选择的程序列表，在列表中选择适当的程序，就会将所选中的文件使用该程序打开。

这种方式只对本次操作有效。也就是说，本次是以用户选择的方式进行打开操作的，但下次双击文件，仍然会以系统默认的方式打开。如果要改变某类型文件的系统默认打开方式，可以按如下步骤进行。

（1）在该文件上右击，在弹出的快捷菜单中选择 Propertis 命令，打开如图 7.6 所示的对话框。

（2）在对话框上选择 Open With 标签，会在列表框中出现一些可用的打开该文件的方式，如图 7.6 所示。

（3）如果要一直以文本编辑器打开该文件，选择 Text Editor 复选框，然后单击 Close 按钮即可。同样如果要以 Emacs22（GTK）打开该文件，选择 Emacs 22（GTK）复选框即可。这样，

以后系统在打开该类型文件时，就会以用户选中的程序打开。

如果在图 7.6 所示的列表中没有要打开该文件的程序，可以单击 Add 按钮选择想要打开文件的程序，然后再将其选中，同样，系统也会以用户另外的选择作为打开文件的方式。

7.4.4　文件管理

文件管理包括文件的复制、剪切、粘贴、创建连接文件等工作。这些内容与 Shell 命令中的功能基本一致，不过在文件管理器中进行这些操作对于初学者来说可能更为简便直观些。由于这部分内容主要是面向初学者，所以介绍得比较详细。对这部分操作熟悉的用户可以跳过此部分内容。

1．复制

文件的复制是指将文件复制一份，而不影响原始文件的一种文件操作。这项操作需要文件的读权限。文件的复制很简单，一般来说有以下三种方法。

- 在选中的文件右击，在弹出的快捷菜单中选择 Copy 命令，即将文件复制在系统剪切板上。
- 或者也可选中文件后，按下 Ctrl+C 组合键，将文件复制。
- 还可以选中文件后，在 Edit 菜单中选择 Copy 命令，完成复制。

2．剪切

文件的剪切与复制很类似，但剪切后，原始文件会被从原始位置删除。这项操作需要该文件的写权限，其操作方法基本类似于复制操作。

- 在选中的文件右击，在弹出的快捷菜单中选择 Cut 菜单项，即将文件剪切下来，放在系统剪切板上。
- 或者也可选中文件后，单击 Ctrl+X 项，将文件剪切。
- 还可以在 Edit 菜单中选择 Cut 菜单项，完成剪切。

3．粘贴

完成了文件复制或剪切后，就可进行粘贴操作。进入到要放置文件的目录，同样，也可以右击，在弹出的快捷菜单中选择 Paste 命令或按 Ctrl+V 将文件粘贴在目录。这个操作需要该目录的写权限。

4．创建链接文件

选择文件，单击鼠标右键，选择 Make Link 就可在当前目录创建一个链接文件。上面介绍了几种主要使用键盘的对文件的操作，其实对文件的这些操作来说，最简单的方法是使用鼠标。

5．文件改名

右击文件，在弹出的快捷菜单中选择“重命名”命令或按下 F2 键，可以看到文件名的位置成为一个文本框，原文件名处于选中状态，按下 Delete 键将原始文件名删除，然后输入新的文件名即可完成文件的改名。

6．删除文件

选择要删除的文件，右击或按下 Delete 键，就可将选中的文件删除。与其他系统类似，文件删除后并没马上就消失，而是先被暂时存放在“回收站”中了。如果需要，还可以将其恢复。

7.4.5　新建目录

目录管理主要包括新建、剪切、复制等操作。除了新建功能外，其余均与文件管理操作相同。这里只介绍新建目录的操作。在要新建目录的目录中右击，选择 Create Folder 命令，目录

中出现一个名为 untitled folder 的目录。同时其文件名位置处是一个文本框，如图 7.7 所示，在其中输入目录名后，就完成了目录的创建工作。

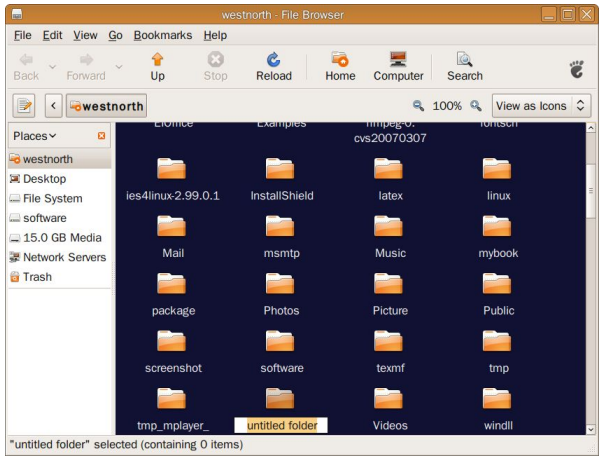


图 7.7　新建目录

7.4.6　给目录、文件增加徽章

有时候，会发现有的文件和目录右上位置有一个徽章图案。这个图案可以很好地区分文件的重要程度、文件种类等信息。要添加这些徽章也很简单，在文件或目录上右击，在弹出的快捷菜单中选择 Properties 命令。在弹出的对话框中选择 Emblems 标签，界面如图 7.8 所示。

在 Emblems 选项卡中列出了一些徽章，选择任一图案左侧的复选框，然后单击 Close 按钮，就可以看到文件的右上侧已经出现了一个漂亮的徽章。

7.4.7　创建压缩文件

右击选择要压缩的文件或目录，在弹出的快捷菜单中选择 Create Archive 命令，便出现如图 7.9 所示的对话框。在 Archive 文本框内输入压缩后的文件名，在后面的下拉框中选择压缩文件类型，然后单击 Create 按钮就可创建一个压缩文件。

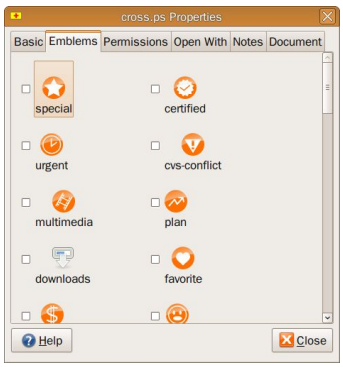


图 7.8　设置文件、目录的徽章

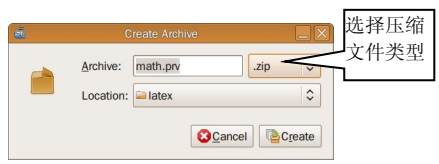


图 7.9　创建压缩文件

7.4.8　可移动存储设备管理

可移动存储设备包括 CD / DVD、以及 USB 设备等。当这些设备插入计算机时，文件管理器会自动识别，并执行相关的操作。图 7.10 显示了 USB 设备与光盘插入电脑后文件管理器的显示状态。可以看到，目录树中已经出现了名为 KODAK 的 USB 设备和名为 Volumelabel 的光盘。

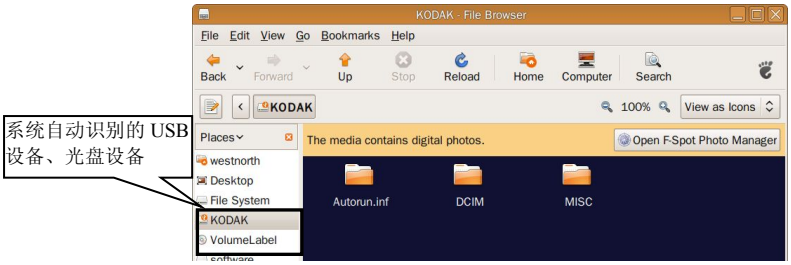


图 7.10　可移动存储设备管理



文件管理器检测到移动设备插入后，会自动提示是否需要用“F-Spot”打开。部分的原因是现在数码相机的流行，用户很大可能插入的是相机存储卡，而用户需要的只是导入 USB 设备中的图片。

7.4.9　权限设置

Linux 中权限可按照组或者用户进行设置。关于用户与组的概念，请参阅本书中关于用户与组的相关章节。本节只介绍如何按组及用户身份设置权限。下面将介绍两种设置权限的方式，一种是在文件管理器中设置，一种是在终端中使用相关的命令来实现。

在文件管理器中设置权限，方法是右击要设置权限的文件或目录，选择 Properties 命令，然后选择 Pemissions 标签，如图 7.11 所示。



图 7.11　权限设置

由图 7.11 可以看出，权限共分为三类，所有者权限、组权限与其他用户权限。而每类权限

又分为两项，目录权限与文件权限。

- 所有者就是文件或目录的创建者，一般来说，所有者的权限与 root 用户是相当的。如图 7.11 所示，所有者对于目录的权限是“Create and delete files”，意思是可以删除和建立文件。文件访问权限是 Read and write，意思是可以读、写文件中内容。
- 组权限是文件所有者所在组其他成员的权限。如图 7.11 所示，所有者在 vboxusers 组，则该组中其他成员的目录权限是可以删除和创建文件，但对于文件而言，只能读而不能写。
- 其他用户权限是除了文件所有者及其所在的组外，其他用户和组登录到这个系统后对于该文件或目录的权限。一般来说，这个权限的设置是最为严格的，如图 7.11 所示只给了其他用户列出文件的权限，他们不能读、写文件。

7.5 chmod 命令设置权限

chmod 是设置权限的命令，功能与 7.4 节所述内容基本一致。不过，相对来说，命令行方式的操作更为强大与灵活些。使用 chmod 设置权限，可以采用两种方式，一种是比较直观的字符方式，另一种是比较灵活的数字掩码方式。这两种方式的效果是一样的，可以根据自己的喜好来选择适当的方式进行各种设置权限的操作。

7.5.1 字符方式设置文件权限

chmod 命令可以识别以字符方式指定的权限。字符方式就是将权限使用其英文单词的首字母来表示，比较方便记忆。如 r 权限代表 read（读）权限、w 代表 write（写）权限。使用字符方式设置权限的命令格式如下：

```
chmod [ugoa...][[+-=][rwx...]...] 文件
```

参数的意义如下。

第一组参数主要是设定用户范围，其意义如下所示。

- u：所有者（user）；
- g：组（group）；
- o：其他用户（other）；
- a：所有用户（all）。

第二组参数用来指定如何设置权限：

- +：增加权限；
- -：取消权限；
- =：设定权限（即不管原先是什么权限都设定为指定的权限）。

第三组参数是权限种类：

- r：（read）可读；
- w：（write）可写；
- x：（execute）可执行。

最后一个参数是要设置权限的文件。

一些命令以及其功能的简单说明列出如下：

```
chmod u+x test.sh          #使文件所有者有运行“test.sh”脚本的权限。
chmod a+w ./*              #使所有用户对当前目录所有文件拥有写的权限。
```

7.5.2　数字掩码方式修改文件权限

使用掩码设置文件权限的命令如下：

```
chmod [mask] 文件...
```

mask 由三个八进制数字组成，该八进制数的百位对应“所有者权限”，而十位对应“组中其他用户权限”，个位对应“其他用户权限”。每个八进制数还应该被分解为一个三位的二进制数。这个三位的二进制数左起第一位代表“读”权限，第二位代表“写”权限，第三位代表“运行”权限。计算表如表 7.1 所示。

表 7.1　权限设置掩码计算表

所有者权限			所有者所在的组权限			其他用户权限		
r（读）	w（写）	x（运行）	r（读）	w（写）	x（运行）	r（读）	w（写）	x（运行）

要设置权限时，只要在相应增加权限的格子里填上 1，不设置权限的格子填上 0。然后每三位一组，合并为一个三位的二进制数，最后将这个三位的二进制数转化为八进制数即可。

如要设置所有者可读、写、运行，而所有者所在组的其他用户只有读与运行的权限，其他用户只有读的权限，就依次在格子里填上相应的数字，如表 7.2 的第三行所示。

表 7.2　权限设置掩码计算示例

所有者权限			所有者所在的组权限			其他用户权限		
r（读）	w（写）	x（运行）	r（读）	w（写）	x（运行）	r（读）	w（写）	x（运行）
1	1	1	1	0	1	1	0	0
7			5			4		

再将这些二进制数分为三组，分别计算其八进制的值。计算时注意其各位的权重值，r 在最高位，其值是 4，w 值是 2，而 x 值是 1。将这些权重分别与各位的值相乘，再加起来就得到该组的位值。如本例中“所有者所在的组权限”的计算，是 1*4+0*2+1*1=5。最后一行所示是计算结果，计算结果是 754。则该命令应该如下：

```
chmod 754 ./*
```

还有一些常用的值，分别如下：

```
#所有者权限为可读、写、运行，其他用户为可读、运行
chmod 755 *
#所有用户权限均为可读、写、运行
chmod 777 *
```

7.6 刻录 CD/DVD 光盘

现在，许多用户的台式机甚至是笔记本上都配置有光盘刻录机。使用光盘刻录机可以方便地将一些不怎么常用却比较重要的数据保存在光盘上。这样做既可以节约硬盘空间，而且也等于给数据做了一个备份，可以在需要时做恢复之用；还有一个额外的好处是方便共享与携带，可以轻松地与他人共享大量的数据。Linux 系统中，可以使用两种方式完成 CD/DVD 刻录工作：使用文件管理器刻录、使用特定软件刻录。

7.6.1 文件管理器刻录

Ubuntu 系统最简单的刻录光盘方式是使用文件管理器。这听起来有点奇怪，文件管理器刻录光盘？是的，Ubuntu 的文件管理器包含有光盘刻录功能，可以轻松完成光盘刻录工作。具体步骤如下。

（1）打开文件管理器，选择菜单“Go”|“CD/DVD Creator”命令，打开图 7.12 所示的窗口。

（2）可以再打开一个文件管理器，然后定位到要刻录的文件，按下鼠标左键，并按住不放，将文件从该窗口中拖出来，拖到刻录窗口中松开鼠标。

（3）要刻录的文件全部放在刻录窗口中后，单击刻录窗口右上侧的 Write to Disk 按钮，弹出 Write to Dise（刻录）对话框。

（4）在 Write to Dise 对话框的 Disc Name 文本框中输入光盘卷标。

（5）单击 Write to Dise 对话框的 Writer 按钮，就会完成刻录。

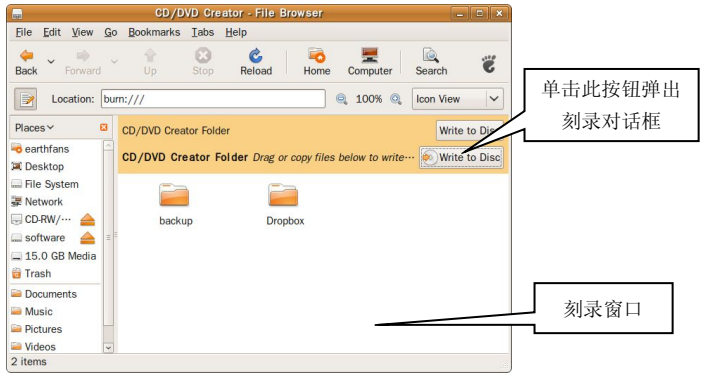


图 7.12 在文件管理器中刻录光盘

7.6.2 专用软件刻录

Ubuntu 自带了一款刻录软件，本节介绍另外一款与之类似的名为 K3b 的软件的使用方法，这两种刻录软件使用方法基本类似。在运行前，要先安装 K3b，在终端中运行命令 sudo apt-get install keb 即可。安装完成后，可在主菜单选择“应用程序”|“音频、视频软件”|“K3b”命令或者直接在终端运行 K3b 即可运行该软件，界面如图 7.13 所示。

K3b 提供了方便的向导功能，向导可以完成音频 CD、数据 CD、数据 DVD、盘片之间复制、刻录 CD 镜像、刻录 DVD 镜像六种工作。这里只介绍数据 CD 与刻录 CD 镜像的功能，

其余功能与此类似，就不再一一介绍了。

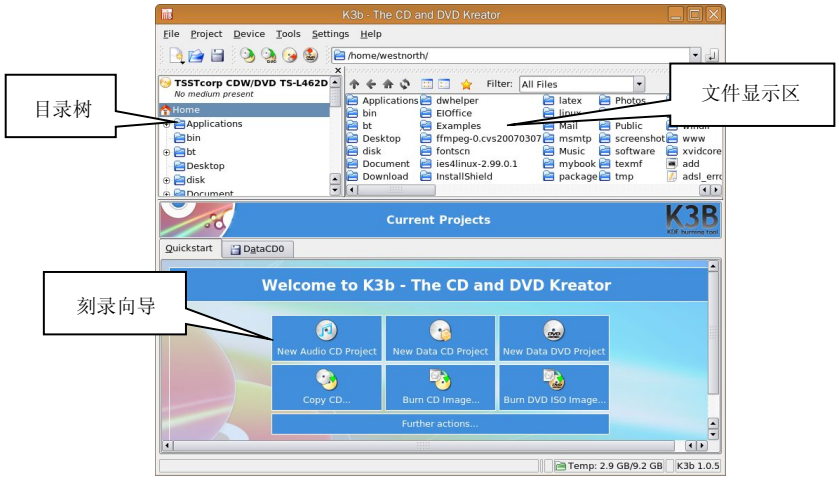


图 7.13　刻录软件 K3b 界面

1．刻录 CD 数据盘

如果要将数据保存到 CD 盘中，可以采用“刻录 CD 数据盘”的向导，具体步骤如下。

（1）单击如图 7.13 所示“刻录向导”区域中的 New Data CD Project 按钮，出现的界面如图 7.14 所示。开始进行 CD 数据盘的刻录工作。

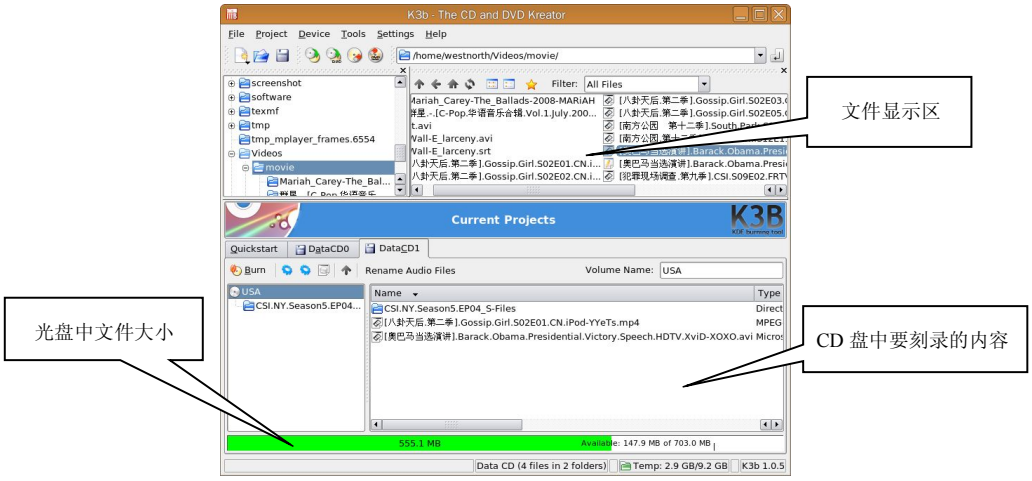


图 7.14　创建 CD 数据盘

（2）从目录树找到要刻录的文件的目录，“文件显示区”应该显示出该目录所包含的文件与目录。

（3）将要刻录的文件从“文件显示区”拖动到“CD 盘中要刻录的内容”区域。这时可以注意到界面最下方有一个绿色的状态条显示了当前文件的大小。如果要刻录的文件大小超过了光盘的最大容量，该进度条就会以红色方式提示。

（4）单击 Burn 按钮，打开如图 7.15 所示的对话框。

（5）选择 Misc 标签，在 Misc 选项卡中将 Multiseesion Mode 下拉框设置为 Start Multisession，

开启多道刻录。



多道刻录意思是本次刻录完成后，如果盘片中有剩余空间，则下次还可以使用这个盘片继续刻录。如果开启该项，光盘一次刻录后，就无法再向里面添加任何内容了。

（6）设置结束后，单击 Burn 按钮就会将所选择的文件刻录到光盘中。

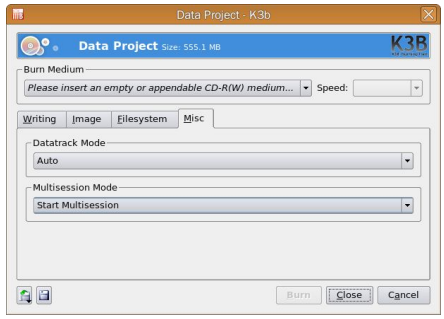


图 7.15　设置刻录选项

2．刻录 CD 镜像

在图 7.13 界面中单击 Burn CD Image 按钮，开始刻录 CD 镜像（ISO 等格式文件）工作，会出现如图 7.16 所示的界面。

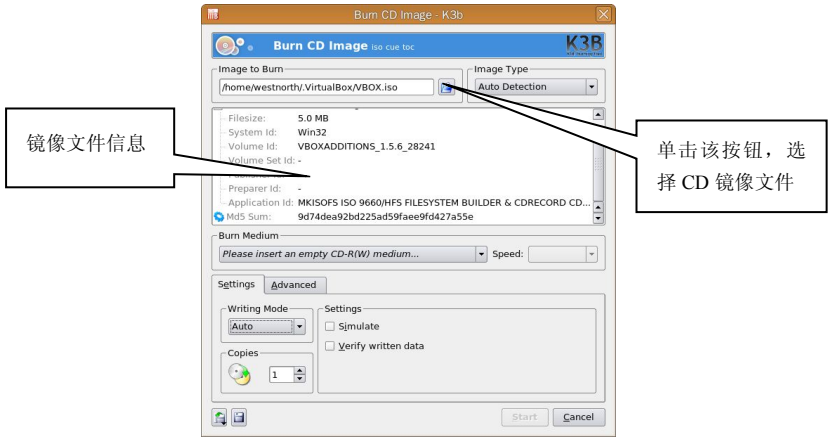


图 7.16　CD 镜像文件刻录

单击图 7.16 所示按钮，选择 CD 镜像文件，会在对话框中间显示镜像文件信息，然后单击“Start”按钮就会开始刻录。

7.7　压缩文件管理

Linux 世界里，各种压缩文件的应用占了很重要的篇幅。大多数源代码都以 tar.gz 为后缀的压缩文件形式发布、系统的帮助手册也是以压缩格式存放的，在使用时才解压。而且很时候在实际应用中，需要将文件压缩以节省空间或节约网络传输的开销。所以，熟悉压缩文件对于

Linux 使用者来说是极其重要的一种技能。本节就介绍一些关于如何压缩文件、目录以及如何解压压缩文件的相关知识。

Linux 中管理有时会将文件打包，就是将多个文件顺序存放在一个文件中，而并不压缩，这种文件一般称为打包文件。这样做的一个原因是备份和还原比较方便，因为有的压缩程序只能处理一个文件。所以，如果要压缩多个文件，就应该先将其打包，然后进行压缩。管理这些打包文件的程序是 tar。在 Linux 中，有许多压缩文件的软件，如 gzip、bzip2 等，这些内容也会在本节中做些介绍。

1．tar 包的处理

解包：tar xvf package.tar。

功能是解开 package.tar 这个包，x 是功能选项，指定功能是解压。而 v 是显示处理过程，f 用来指定包名。

打包：tar cvf package.tar directory。

功能是将 direcotry 目录里的所有文件与目录全部压缩（准确地说是打包）成一个名为 package.tar 的文件中。参数的意义可参照上一条命令的说明。

2．gz 压缩文件的处理

解压：gunzip -d zipfile.gz。

功能是将 gz 文件解压缩也可用 gzip 来实现，这两个命令除了程序名不同外，功能均是一样的。

压缩：gzip FileName。

功能是将文件 FileName 压缩到本地同名的.gz 文件中。

3．tar.gz 和 tgz 文件的处理

解压：tar zxvf tarfile.tar.gz。

功能是将 tarfile.tar.gz 解压到当前目录。

压缩：tar zcvf tarfile.tar.gz orderFolder。

功能是将 orderFolder 目录中的文件压缩到文件 tarfile.tar.gz。

4．bz2 文件的处理

解压：bzip2 -d bzfile.bz2。

功能是将 bzfile.bz2 解压到当前目录。

压缩：bzip2 -z bzfile。

功能是将 bzfile 压缩到同名文件。

5．tar.bz2 文件的处理

解压：tar jxvf tarbzfile.tar.bz2。

功能是解压文件。

压缩：tar jcvf tarbzfile.tar.bz2 orderFolder。

功能是压缩文件。

6．bz 文件的处理

解压：bzip2 -d bzfile.bz。

功能是解压文件。

7．tar.bz 文件的处理

解压：tar jxvf File.tar.bz。

8．tar.Z 文件的处理

解压：tar Zxvf File.tar.Z。

压缩：tar Zcvf File.tar.Z Folder。

9．zip 文件的处理

解压：unzip File.zip。

压缩：zip File.zip Folder。

10．rar 文件的处理

解压：rar a File.rar。

压缩：rar e File.rar。

11．deb 文件的处理

解包：arp File.deb debfile.tar.gz | tar zxf -。

12．Z 文件的处理

解压：uncompress all.Z。

7.8 小结

本章主要介绍了 Linux 文件系统管理的相关知识。首先介绍了 Linux 文件管理的基础知识——目录结构；然后介绍了加载其他文件系统的方法；接着在文件管理器中，介绍了对初级用户比较实用的入门级操作；最后介绍了各种压缩文件的解压及压缩的方法。本章的重点内容是加载文件系统，以及压缩文件管理这两节。相对来说，开机自动加载、权限设置的内容对初级用户来说可能比较困难。文件管理器的使用是每个计算机用户都应该掌握的基础知识，而且其操作比较简单，多加练习应该是很容易掌握的。

第 8 章　软件包管理

其实 Linux 中所谓软件包管理的工作就是安装软件与卸载不再需要的软件。原来的 Linux 发行版最让普通用户诟病的就是缺乏简单易行的软件包管理功能。时至今日，Ubuntu 提供了多样化的软件安装服务，如简单易行的安装/删除命令、强大的新立得软件管理器，以及各种命令行的方式等。本章将介绍这些内容。

8.1　安装前的准备工作

Ubuntu 使用了一种称为软件源的技术来提供系统中所需的软件资源。所谓的源就是一台包含了许多 Ubuntu 软件的服务器。这台服务器采用了特殊的技术构建而成，可以识别网络中 Ubuntu 用户的安装软件命令，然后自动将用户所需要的软件传送到用户计算机中进行安装。用户在使用源进行安装前，需要进行一些必要的设置，以达到最佳性能。

8.1.1　源服务器简介

对于普通用户而言，只要在系统中一个叫做源列表的文件中添加源服务器的名称，就可以安装该源服务器中的软件，而且使用源服务器安装软件的过程也很简单。提供“源”服务的服务器又可分为官方服务器、镜像服务器，以及其他服务器。

1．官方服务器

官方服务器对于版权要求极其严格，稍有版权问题的软件均不会收录。所以相对来说，普通的应用软件收录得可能不是很全面。但对于构建系统所必需的软件，以及 GNU 软件，收录则是极其完整的。

2．镜像服务器

如果全世界所有用户均到官方的源服务器下载软件，服务器很可能会无法应付如此大的数据流量而宕机。与此同时，一些教育机构出于教学与研究的需要，也需要建立自己的源服务器。所以，Ubuntu 提供了下载其源中所有的软件制作源服务器的功能。

只要将官方源中软件全部下载到一个计算机中，然后经过特殊的设置，就可以使该计算机成为源服务器，供其用户使用。这个由 Ubuntu 源服务器全盘复制得到的服务器称为镜像服务器。如果可以提供一定的带宽和稳定的服务器，经过 Ubuntu 的审核后，就可以让镜像服务器作为官方服务器的镜像，供全球的用户下载之用。

3．其他服务器

还有一种源服务器，其没有得到 Ubuntu 的正式认可，但该源服务器提供了一些特别的软件（如 MediaUbuntu 提供了音、视频解码器，而 WineHQ 提供了 Wine 的最新软件包等）。将这些源加入到源列表中，就可以像安装其他软件一样，安装这些特定的软件了。

8.1.2 添加源列表

所谓的源列表，就是源服务器的列表。将这些列表加入源列表文件后，就可以安装该服务器中的软件了。一般在源列表中要加入官方源（官方服务器提供的源），这是保证系统所需的核心类库的完整性；还要加入一些本地源（镜像服务器提供的源），这是为了保证下载软件的速度，这是因为一般本地源的速度要远高于官方源。

系统在扫描源列表时，会将源服务器中软件的版本生成一个数据库。要安装一个软件时，会自动从数据库中搜索该软件最新的版本。如果各源服务器中软件版本一致，则按照服务器在源列表中的顺序进行安装。所以在源列表文件添加服务器源时，应将本地源服务器（速度快）添加在源列表文件的前面。

源列表文件是/etc/apt/sources.list。在向其添加源列表时，应该先将原始内容备份一份，这是为了在添加源列表发生错误后恢复使用。一般来说，在修改重要的系统配置文件前进行备份是一个良好的操作习惯。

1．备份源列表文件

在终端中使用以下命令生成一个名为 sources.list.backup 的备份文件：

```
sudo cp /etc/apt/sources.list /etc/apt/sources.list.backup
```

2．添加源列表

同样在终端中使用以下命令就可打开源列表文件：

```
sudo gedit /etc/apt/sources.list
```

打开文件后，将其中原始文本全部删除（这是因为 Ubuntu 安装时默认设置的源一般 gf 位于欧洲或美洲，对于我国用户来说，其速度很慢），然后开始添加自己的源。首先应该在其中添加本地源，下面列出了各种线路（电信、联通、校园网）用户可以使用的本地源。

下面源列表中以说明文字隔开的一组文字为一个源服务器。可以看到每一个服务器都有许多行内容，这是因为服务器中软件有许多种类，每行代表一种类型的软件。在添加源时要将该服务器所有的行均添加进去。可参照自己的网络情况，添加其中任意一个服务器。



添加多个服务器是没有什么意义的，只会增加刷新软件库的时间。

```
#上海交大源服务器（教育网用户可以选择此源）
deb http://ftp.sjtu.edu.cn/ubuntu/ hardy main multiverse restricted universe
deb http://ftp.sjtu.edu.cn/ubuntu/ hardy-backports main multiverse restricted universe
deb http://ftp.sjtu.edu.cn/ubuntu/ hardy-proposed main multiverse restricted universe
deb http://ftp.sjtu.edu.cn/ubuntu/ hardy-security main multiverse restricted universe
deb http://ftp.sjtu.edu.cn/ubuntu/ hardy-updates main multiverse restricted universe
deb-src http://ftp.sjtu.edu.cn/ubuntu/ hardy main multiverse restricted universe
deb-src http://ftp.sjtu.edu.cn/ubuntu/ hardy-backports main multiverse restricted universe
```

```
    deb-src http://ftp.sjtu.edu.cn/ubuntu/ hardy-proposed main multiverse
restricted universe
    deb-src http://ftp.sjtu.edu.cn/ubuntu/ hardy-security main multiverse
restricted universe
    deb-src http://ftp.sjtu.edu.cn/ubuntu/ hardy-updates main multiverse
restricted universe
    #中国台湾大学源服务器（网通用户可以使用此源，其源比较完整）
    deb http://ubuntu.csie.ntu.edu.tw/ubuntu/ hardy main restricted universe
multiverse
    deb-src http://ubuntu.csie.ntu.edu.tw/ubuntu/ hardy main restricted
universe multiverse
    deb http://ubuntu.csie.ntu.edu.tw/ubuntu/ hardy-updates main restricted
universe multiverse
    deb-src http://ubuntu.csie.ntu.edu.tw/ubuntu/ hardy-updates main
restricted universe multiverse
    deb http://ubuntu.csie.ntu.edu.tw/ubuntu/ hardy-backports main restricted
universe multiverse
    deb-src http://ubuntu.csie.ntu.edu.tw/ubuntu/ hardy-backports main
restricted universe multiverse
    deb http://ubuntu.csie.ntu.edu.tw/ubuntu/ hardy-security main restricted
universe multiverse
    deb-src http://ubuntu.csie.ntu.edu.tw/ubuntu/ hardy-security main
restricted universe multiverse
    deb http://ubuntu.csie.ntu.edu.tw/ubuntu/ hardy-proposed main multiverse
restricted universe
    deb-src http://ubuntu.csie.ntu.edu.tw/ubuntu/ hardy-proposed main
restricted universe multiverse
    #上海的一个服务器（电信用户可以使用此源）
    deb http://mirror.rootguide.org/ubuntu/ hardy main restricted universe
multiverse
    deb-src http://mirror.rootguide.org/ubuntu/ hardy main restricted universe
multiverse
    deb http://mirror.rootguide.org/ubuntu/ hardy-updates main restricted
universe multiverse
    deb-src http://mirror.rootguide.org/ubuntu/ hardy-updates main restricted
universe multiverse
    deb http://mirror.rootguide.org/ubuntu/ hardy-backports main restricted
universe multiverse
    deb-src http://mirror.rootguide.org/ubuntu/ hardy-backports main
restricted universe multiverse
    deb http://mirror.rootguide.org/ubuntu/ hardy-security main restricted
universe multiverse
    deb-src http://mirror.rootguide.org/ubuntu/ hardy-security main restricted
universe multiverse
```

```
    deb http://mirror.rootguide.org/ubuntu/ hardy-proposed main multiverse
restricted universe
    deb-src http://mirror.rootguide.org/ubuntu/ hardy-proposed main restricted
universe multiverse
```

3．添加官方源

添加了这些本地源后，有时还会出现无法找到某些软件的问题，这就需要在源列表中添加官方源来解决这个问题。同样，以下这些官方源列表任意添加一个即可。

```
    #中国台湾的官方源服务器，速度不错
    deb http://tw.archive.ubuntu.com/ubuntu hardy main restricted universe
multiverse
    deb http://tw.archive.ubuntu.com/ubuntu hardy-security main restricted
universe multiverse
    deb http://tw.archive.ubuntu.com/ubuntu hardy-updates main restricted
universe multiverse
    deb http://tw.archive.ubuntu.com/ubuntu hardy-backports main restricted
universe multiverse
    deb http://tw.archive.ubuntu.com/ubuntu hardy-proposed main restricted
universe multiverse
    deb-src http://tw.archive.ubuntu.com/ubuntu hardy main restricted universe
multiverse
    deb-src http://tw.archive.ubuntu.com/ubuntu hardy-security main restricted
universe multiverse
    deb-src http://tw.archive.ubuntu.com/ubuntu hardy-updates main restricted
universe multiverse
    deb-src http://tw.archive.ubuntu.com/ubuntu hardy-backports main
restricted universe multiverse
    deb-src http://tw.archive.ubuntu.com/ubuntu hardy-proposed main restricted
universe multiverse
    #位于欧洲的官方源，电信、网通用户可使用
    deb http://archive.ubuntu.com/ubuntu/ hardy main restricted universe
multiverse
    deb http://archive.ubuntu.com/ubuntu/ hardy-security main restricted
universe multiverse
    deb http://archive.ubuntu.com/ubuntu/ hardy-updates main restricted
universe multiverse
    deb http://archive.ubuntu.com/ubuntu/ hardy-proposed main restricted
universe multiverse
    deb http://archive.ubuntu.com/ubuntu/ hardy-backports main restricted
universe multiverse
    deb-src http://archive.ubuntu.com/ubuntu/ hardy main restricted universe
multiverse
    deb-src http://archive.ubuntu.com/ubuntu/ hardy-security main restricted
universe multiverse
    deb-src http://archive.ubuntu.com/ubuntu/ hardy-updates main restricted
universe multiverse
    deb-src http://archive.ubuntu.com/ubuntu/ hardy-proposed main restricted
universe multiverse
```

```
deb-src http://archive.ubuntu.com/ubuntu/ hardy-backports main restricted
universe multiverse
```

4．添加光盘作为源列表

如果拥有 Ubuntu 的 DVD 安装盘，还可以将 DVD 盘也加入源列表中。由于 DVD 盘中有系统核心类库及一些常用的软件集合，使用 DVD 做源后可以节省许多下载的时间。在命令行添加光盘的方法是将光盘放入光驱中，然后运行如下命令：

```
sudo apt-cdrom add
```

几分钟后，就会将光盘中的软件信息添加到软件源中，如果在安装中需要光盘上的某个软件，在安装过程中会给出提示，按提示放入光盘即可。

5．刷新源列表

源列表发生任何更新，如添加或删除源列表、添加光盘作为源等，均需要刷新源列表，以获得最新的软件信息。无论修改的是不是网络中的源，在刷新源列表时均需要与网络链接。刷新源列表是很简单的，只需要在终端运行如下命令即可：

```
sudo apt-get update
```

执行此命令后，耐心等待一会，就会完成源列表的刷新。完成后，就可以进行软件的安装了。

8.2　使用主菜单中“添加/删除”命令管理软件

Ubuntu 软件源软件丰富并且没有版权问题，而且安装方式操作简单、功能强大，对于普通用户来说也很容易上手。在 Ubuntu 的软件源里，有几万款软件，这些软件涵盖了办公、游戏、教育、影音处理等各个方面，安装时只需简单单击鼠标即可。完成 8.1 节的源列表的相关设置后，就可以正式开始安装过程了。本节介绍一种最简单的安装方式——使用“添加/删除”命令管理软件。

8.2.1　“添加/删除”命令界面

该软件名为 gnome-app-install，可以在终端中运行该软件。但最简单的方法是依次单击主菜单“应用程序”|“添加/删除”命令，就可打开该软件。软件界面如图 8.1 所示。



图 8.1　“添加 / 删除”软件界面

“软件分类”列表框中列出了可供选择的软件种类，包括附件，教育、游戏、图像、网络、办公、编程、声音与视频、系统软件等几种。下面是这些软件种类中包含的软件简单介绍。

- “附件”类中包括了常用的轻量级文本编辑器、文书浏览器、压缩文件管理器等软件。
- “教育”类中包含了一些用于儿童教育辅助的软件，以及一些科学软件，如 scilab 等。
- “图像”类中包含了用于查看图像、漫画或处理图像，以及三维建模的软件。
- “网络”类中包含了网络浏览器，以及网络中需要的一些工具软件。
- “系统”类中包含了系统设置中使用的软件，包括无线网卡的驱动，其名字为 Ndiswrapper。

8.2.2 安装软件

在“添加/删除”命令中安装软件也是很容易的。而且该程序已经将软件分门别类地排列好，要安装某些特定种类的软件，直接到该类软件中寻找即可，可以大大节约寻找软件的时间。在浏览该分类的过程中，说不定还会发现一些原来不知道的软件。具体安装过程可以参照如下步骤进行。

（1）在图 8.1 所示左侧的“软件分类”列表框中单击感兴趣的分类，如 Education（教育），在右侧“软件列表”列表框中就会出现教育类软件的列表，可以在其中寻找想要安装的软件。

如果知道软件的名字，要在软件库中搜索该软件，就在“搜索”文本框中输入软件名，按下 Enter 键就可在“软件列表”中得到与输入文字匹配的软件。

（2）找到了需要安装的软件，就在“软件列表”列表框中单击该软件前的复选框，将其勾选，再单击“应用改变”按钮就开始了软件的安装。

（3）软件安装第一步是应用更改，也就是确认安装。如图 8.2 所示，在“添加”列表框中列出了本次要安装的所有软件。如果显示出的软件就是要安装的，就单击“应用”按钮，开始下载。

（4）下载软件包的过程如图 8.3 所示。由于 Ubuntu 的软件源一般均放置在网络服务器上，所以在安装过程中需要保证联网以下载软件。

（5）下载完成后，会自动来到软件安装对话框。这一步才将下载的软件安装在系统中。界面如图 8.4 所示。等待一会儿，会出现“安装成功”的提示信息，说明安装已经顺利完成。单击“关闭”按钮完成安装。



图 8.2 应用更改

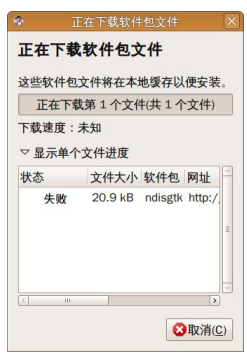


图 8.3 下载软件包



图 8.4 软件安装

8.2.3　卸载软件

与安装软件类似，在“添加/删除”命令中卸载软件也是很容易的。卸载的过程就是定位到特定的程序，然后取消选中状态，就可以完成了。卸载软件的具体步骤如下所示：

（1）定位到要卸载的软件；

（2）单击该软件前带有对号的小方框，取消其选中状态，意思是卸载软件；

（3）单击“应用更改”按钮即可开始卸载过程，具体过程与安装软件类似，可参照提示进行操作即可完成卸载软件的工作。

可以看到，使用这种方式安装、卸载软件的过程是比较简单的，对于普通用户来说也可以轻松上手。值得一提的是，图 8.1 所示列表中的软件极其丰富。一般来说，可以满足绝大多数用户的需求。

8.3　使用新立得软件管理器管理软件

“安装 / 删除”命令虽然简单，但其还是有些弱点的，如对于系统软件的依赖问题解决得不太好，如果遇到比较复杂的软件依赖，就会提示说“本次安装无法继续，建议使用新立得软件管理器进行安装”。新立得软件管理器是 Ubuntu 用户最基本的软件安装方式。“添加/删除”命令只可安装应用软件，而新立得软件管理器不只可以安装应用软件，也可安装系统底层的库文件、程序的源代码，还可以自动解决软件库的依赖问题，以及卸载已安装的程序。可以说，在新立得软件管理器中可以完成所有的软件管理任务。

8.3.1　新立得软件管理器界面

打开新立得软件管理器的方法是，在终端中输入 sudo synaptic 或依次选择“系统”|“系统管理”|“新立得软件管理器”命令，新立得软件管理器的界面如图 8.5 所示。



图 8.5　新立得软件管理器界面

可以看出，其界面的基本结构与“安装 / 删除”命令类似，界面也分为三部分，左侧列表栏列出了软件资源分类，右侧上端列表栏列出了软件名，而下端则是软件的简介。

看图中最下端的状态栏，显示的内容是“共列出 25171 个软件包，已安装 1601 个，已坏损 0 个，将安装/升级 0 个软件包，将删除 0 个软件包”。说明新立得软件管理器管理的软件包有 25171 个，还有一些其他与软件安装相关的信息。对于软件管理来说，状态栏是很重要的一栏。

1．软件分类

“软件分类”列表框中以分类方式列出了各种软件。新立得软件管理器提供了更多的分类来方便地浏览软件信息。

- 图 8.5 中选择的分类方式是“组别”，意思是按照软件的功能来分类显示。
- 选择“状态”选项，可以按软件的安装状态不同，如“已安装”、“未安装”、“已安装”来查看软件包信息。
- 选择“源自”选项，可以按照源服务器不同以及源服务器的频道不同来分类显示软件包信息。
- 选择“自定义过滤器”选项，可以按照用户自定义的方式，如设置“搜索条件”等来显示软件。
- 选择“搜索结果”选项，则按照用户的搜索结果显示符合搜索条件的软件。

2．软件列表

软件列表中显示了符合“软件列表”设置的软件，这是软件操作的主窗口，要对软件进行操作，右击该软件后，会出现相关的设置菜单，如图 8.6 所示。

- 取消标记：取消对该软件做的所有标记，如安装标记、删除标记等。
- 标记以便安装：要安装此软件，就标记该项。
- 标记以便重新安装：如果软件已经安装，但由于某些原因（如使用中出现问题），需要重新安装时选择该项。
- 标记以便升级：如果该项在源数据库中更新的版本，会用新版本替换当前的旧版本。
- 标记以便删除：删除该软件。
- 标记以便彻底删除：删除该软件并删除该软件的配置文件。
- 属性：查看该软件包的属性，如软件的版本、依赖关系、将该软件包含的文件安装在系统的什么位置等信息，如图 8.7 所示。



图 8.6　软件设置菜单



图 8.7　查看软件属性

- 标记推荐的以便安装：选择某项软件包时，系统会根据软件的相关程度，推荐可以增强

该项软件功能的一些附件。

- 标记建议的以便安装：系统建议安装的一些软件包，一般来说对于某些特殊需要的用户可能有用。



要安装所选择的软件，最简单的方法是双击该软件包，即将其标记为安装。

3．软件简介

图 8.5 所示界面中“软件简介”文本框显示的内容是“软件列表”列表框中所选中软件的描述，一般是软件的功能特点等，有时也会有软件历史等的介绍。

8.3.2　软件源管理

在新立得软件管理器中，也可以方便地进行软件源的管理。软件源的管理包括添加软件源，以及刷新软件源等内容。

1．添加软件源

在新立得软件管理器中，添加软件源是一项很琐碎的工作，远不如在文本界面中添加软件源快捷高效，不过，对于初学者来说可能比较容易接受。下面是添加软件源的具体步骤。

（1）选择新立得软件管理器菜单“设置”|“软件库”命令，出现图 8.8 所示对话框。

（2）选择“第三方软件”标签，进入该标签页。

（3）单击对话框中“添加”按钮，出现如图 8.9 所示的对话框。在其中“APT 行”文本框中输入要添加的软件源。注意这里输入的字符必须符合源特定的格式，具体格式可以参见 8.1.1 节相关内容。



图 8.8　增加软件源



图 8.9　输入软件源字符串

如可在“APT 行”文本框中添加字符串“deb http://ftp.sjtu.edu.cn/ubuntu/ hardy main multiverse restricted universe”。其中：

- “deb”是源类型，意思是该源提供的是 Deb 包；
- “http://ftp.sjtu.edu.cn/ubuntu/”是服务器地址；
- “hardy”是 Ubuntu 版本；
- “main multiverse restricted universe”是软件的频道。

参照此格式输入源字符串后，单击“添加源”按钮就完成添加源列表的工作。不过，8.1.2

节提到，每个源服务器均有多个软件分类，需要将这些分类全部添加到源列表中。所以，这个过程可能要重复许多次才能完整地添加一个源服务器。



在如图 8.8 所示的界面中还可以完成添加光盘作为源的工作。方法是单击“添加光驱”按钮，插入 Ubuntu DVD，单击“确定”按钮就可完成添加 DVD 作为源的工作。

2．刷新软件包信息

选择新立得软件管理器中菜单“编辑”|“重新载入软件包信息”命令，或者简单地在新立得软件管理器中按下 Ctrl+R 键即可自动刷新软件源中的软件包信息。刷新软件包信息如图 8.10 所示。

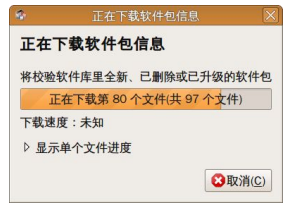


图 8.10　刷新软件包信息

8.3.3　软件管理

当然，新立得软件管理器最主要的功能还是进行软件管理，包括搜索软件、安装软件、解决软件依赖问题等。掌握了这些功能，会给 Ubuntu 的使用带来极大的便利。因为熟练这些操作后，就可随时安装自己想要的软件，从很大程度上增加系统的功能，完成各种任务。下面分别介绍软件管理的各项内容，包括搜索软件、安装软件、卸载软件等。

1．搜索软件

在 8.3.1 节相关内容中已经介绍了如何通过软件分类定位软件的方法，现在再来介绍一种更为灵活的定位软件的方式，就是通过搜索定位软件。新立得软件管理器提供了很强的搜索功能，可以根据软件名、软件的功能描述等信息来搜索软件。

使用新立得软件管理器强大的搜索功能方法很简单，直接单击工具栏中“搜索”按钮即可出现如图 8.11 所示的对话框。

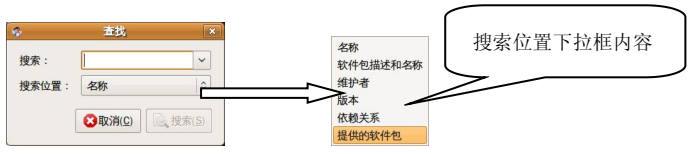


图 8.11　搜索软件

单击“搜索位置”下拉框，会出现搜索选项，最常用的是前两项，其意义如下所示。

- 名称：只在软件名称中搜索匹配内容。如果知道软件名或软件名的一部分，可以使用本项。本项搜索出的结果最为精确，但数量也比较少，如果名字搞错，就会导致搜索不到结果。
- 软件包描述信息和名称：在软件名称与软件的描述（就是图 8.2“软件简介”）中搜索匹配内容。由于搜索时对“软件简介”的文本也进行搜索，因此本项搜索出来的结果最多，但是有可能会搜索到许多与输入内容不相关的软件，需要用户自己从“软件简介”中判断到底哪一款才是自己想要的。两种搜索方式搜索同一个字符串 maxima 的结果对比如图 8.12 与图 8.13 所示。

从图 8.12 与图 8.13 来看，显然使用“名称”选项搜索到的结果更符合要求，列在第一项的就是 maxima 软件。不过，从状态栏可以看出，使用“名称”选项只搜索到 8 个结果，而使

用“软件包描述信息和名称”搜索到了 22 个结果。



图 8.12　使用“名称”选项的搜索结果



图 8.13　使用“软件包描述信息和名称”选项搜索的结果

2．标记软件状态

定位到所要安装的软件后，在该软件上双击或右击选择“标记以便安装”命令，就将其设置为“安装”状态。这是因为新立得软件管理器在安装软件时，只安装标记为“安装”状态的软件，所以要安装一个软件，需先将其标记为“安装”状态。

3．安装软件

选择好要安装的软件后，就可单击工具栏带有绿色对号的“应用”按钮开始安装。

安装前系统会检查软件的依赖情况。如果发现依赖关系没有满足，会自动标记安装该软件所必需的其他软件，同时提示用户是否将这些必需的软件也标记为“安装”状态。这时应该选择“标记”，系统标记那些软件为“安装”状态，同时退回主界面。再单击“应用”按钮，系统就会安装所有标记为“安装”的软件。

4．解决软件问题

由于错误的操作或者其他原因，有时会导致软件无法正常使用。一般可以尝试重新安装该软件来解决这个问题。方法是在如图 8.5 所示“软件列表”中定位到该软件，然后右击该软件，在弹出的快捷菜单中选择“标记以便重新安装”命令，再单击主界面中工具栏“应用”按钮就可完成软件的重新安装工作。

如果由于强行安装某些不兼容的版本或卸载了软件需要的类库，出现软件依赖的问题，可以依次打开新立得软件管理器的菜单“编辑”|“修复（依赖关系损毁的）软件”命令，系统经过一段时间的修复后，会在状态栏显示“软件依赖关系修复完毕”。经过此操作，一般情况下软件依赖关系的问题会得到解决。

8.4　安装其他格式的软件包

虽然新立得软件管理器也就是源服务器已经提供了种类极多、性能优良的软件，但是总会有这些软件还无法满足需求的时候。如果需要的软件在新立得软件管理器中找不到也没关系，由于 Linux 是符合 POSIX 标准的操作系统，网络中有很多可运行在 Linux 平台的软件供下载。只要找到这些软件，下载到电脑上，进行安装，然后就可正常使用了。

8.4.1 安装 Deb 包格式的软件

网络上可供下载的软件中，应该优先选择 Deb 包格式的软件。这些 Deb 包是专门为 Ubuntu 系统编译的，Ubuntu 用户使用极为方便。而且该包在安装中会自动处理软件依赖，用户不用再处理复杂的依赖关系，这样会降低软件安装中出现问题的几率。

可以在 http://www.getdeb.net 网站下载 Deb 软件。该网站汇集了许多 Ubuntu 平台的 Deb 软件，资源很丰富，网站主页如图 8.14 所示。

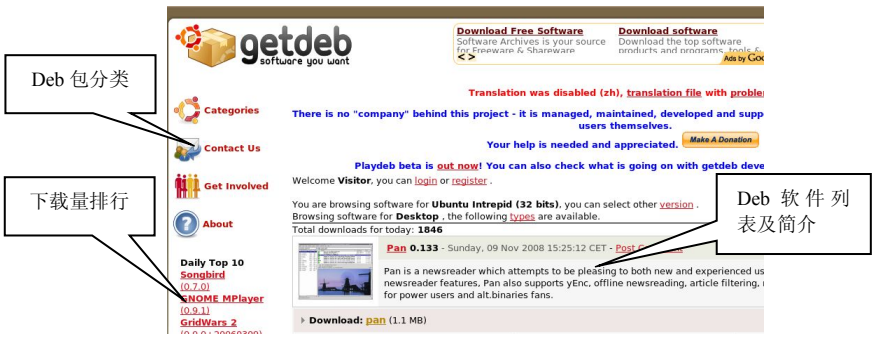


图 8.14　getdeb 网站主页

选择“Deb 包分类（Categories）”选项，进入 Deb 包分类页面，就可以按照软件分类寻找想要的软件，或者可在下载量排行中查看本日下载量最多的几款软件。

在主页中网站还会推荐几款软件，列在“Deb 软件列表及简介”区域中。该区域中不但列出了软件名称，还给出了软件的简介、软件的运行截图以及软件的大小等信息，如图 8.15 所示。



图 8.15　从 getdeb.net 中下载软件

下面以 fracplanet 这款小软件为例介绍下载及安装的过程，具体步骤如下。

（1）单击如图 8.15 所示 Download 字符后面的链接即可将该 Deb 包下载到用户的计算机中。

（2）从文件管理器中找到下载的 Deb 包，双击该包开始安装。

一般情况下，双击后就会开始安装。但本例选择的 fracplanet 其依赖关系并未得到满足，双击后会出现如图 8.16 所示的提示：

在“状态”提示中文字的意思是 libboost-program-options 包没有安装。

（3）打开新立得软件管理器，单击“搜索”工具按钮，然后参照本章 8.3.3 节第 1 项“搜索软件”中内容，选择“名称”选项，搜索名为 libboost-program-options 的包。

（4）将搜索到的包标记为“安装”状态，单击“应用”工具按钮将搜索到的两项结果全部安装。

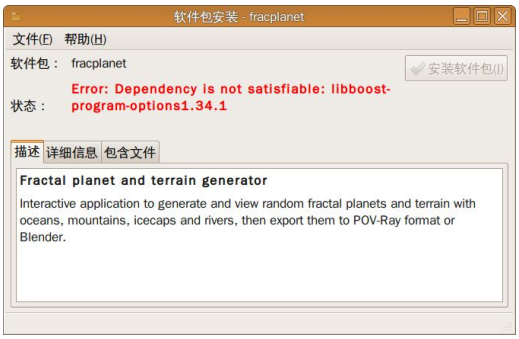


图 8.16　依赖关系没有满足时安装 deb 包的提示信息

（5）安装好依赖关系后，再重新双击该 Deb 包，状态栏已经没有红色字体的错误提示，单击“安装软件包”按钮即可完成此安装。

其实，一般的 Deb 包安装是极其简单的，大多数情况下会自动处理依赖问题。如果出现像上例的无法自动处理依赖问题的情况，只需按提示将所未满足的依赖关系（一般是库文件）安装后就可正常安装 Deb 包了。官方的 Deb 包下载地址是 http://packages.ubuntu.com/。

8.4.2　编译、安装提供源代码的软件

用源码安装应该是安装方式中的高级方式，这种方式是指下载软件的源代码，自己进行编译、安装的过程。相对于使用“新立得”和 Deb 包来说，源码安装可以调整的参数多，可以按照自己的需求包含或不包含相应的功能，比较灵活；而且在本机上编译的软件，编译器会自动根据机器的配置采用相应的优化措施，会让软件发挥最好的性能；还可以控制安装的位置，便于软件的管理。

但是，这种方式比较复杂，如果对于编译参数理解不够、参数使用失误，可能会造成编译出的软件无法使用的情况；而且采用源码安装，遇到的依赖情况会更复杂，要根据提示信息灵活处理。这些对于没有经验的初学者来说是个很大的挑战。

一般情况下，开始接触 Linux 的时候，比较小的软件可以考虑自己编译。比较复杂的软件最好使用别人编译好的 Deb 包。而有经验的开发人员就可以自己编译软件以达到最好的性能了。下面简单介绍一下编译源码安装软件的过程。

1．构建编译环境

编译源代码前，要先构建编译环境。由于 Ubuntu 是面向普通用户的，系统中并未默认安装编译工具，需要自己安装。安装命令如下：

```
sudo apt-get install build-essential autoconf automake1.9 cvs subversion
```

当然这些软件在“新立得”中也可安装，但效率太低。所以推荐在终端中运行此命令。

2．下载源码

以一个简单的文字界面网络浏览器 lynx 为例，介绍编译安装的过程。可以使用如下命令获得源码：

```
wget http://ftp.de.debian.org/debian/pool/main/l/lynx/lynx_2.8.5.orig.tar. gz
```

lynx 是一款文本界面的网络浏览器，其特点是浏览速度快，其运行界面如图 8.17 所示。

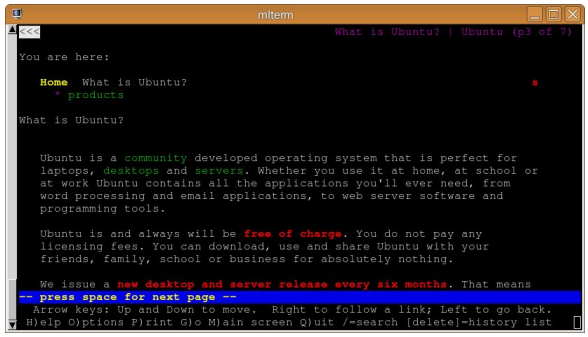


图 8.17　lynx 文本方式浏览网页

3．解压源码

由于下载得到的文件是压缩文件，在使用前须进行解压工作。解压源码的命令如下：

```
tar xzvf  lynx_2.8.5.orig.tar.gz
cd lynx2-8-5                    #更改当前目录到代码所在的目录
```

4．编译、安装

下面的三步是通用的源代码编译命令，大多数软件源代码均可以简单地通过这三条命令来进行编译安装：

```
#如果要实现特定的功能，应该使用“./configure --help”查看可用的安装选项
#然后将需要的选项加入其中，如本例可用“./configure --with-x”
./configure
make
sudo make install
```

5．运行

在终端中输入 lynx 就可以使用 lynx 进行网络浏览了，运行界面如图 8.17 所示。

8.5　软件包管理命令

图形界面的软件安装方式虽然简单，但是相对来说其效率并不高。而且从本质上来说，这些图形界面的软件是构建在一系列命令的基础上的，所以熟练掌握这些命令行方式的软件安装对于各种环境中（如远程登录）软件包管理有着重要的意义。

8.5.1　apt 系列

apt 命令是 Ubuntu 系统中管理软件包的主要命令，可以完成许多复杂的软件管理工作。apt 是一个复杂而且庞大的系统，有 apt-ge、apt-cache 等命令。



本节所示各命令中“package”不是命令的一部分，而是用户定义的任意软件包名。

1．apt-get 命令

apt-get 是最主要的软件包管理命令，可以完成软件的安装、卸载，以至于系统升级的功能。这也是软件包管理中最常用的命令，应该熟练掌握。其使用格式如下：

```
apt-get [选项] 命令
```

选项说明如下所示。

- -h：显示帮助文档。
- -qq：只显示错误信息，不显示其他信息。
- -d：仅从源服务器上下载软件包，但是并不进行实际的安装。
- -s：模拟执行命令，并不实际运行。
- -y：对于所有提示“yes/no”的操作，均选择肯定的回答。这个选项应该慎用，因为这个选项可能会安装用户不需要的软件或者卸载系统中重要的库文件。
- -f：强制安装软件包，即使没有通过完整性验证。
- -m：当缺少某些包时，仍然继续安装。
- -u：显示升级了的软件包列表。
- -b：下载源码包后，自动编译，并生成相应的软件包。

下面是 apt 的一些管理命令列表。

- sudo apt-get update：刷新系统中的软件信息，在源列表文件“/etc/apt/sources.list”被修改后使用。
- sudo apt-get install package：安装软件，下载的 deb 包保存在“/var/cache/apt/archives/”目录。
- sudo apt-get install --reinstall package：强制重新安装软件。
- sudo apt-get -f install：修复安装中出现的错误。
- sudo apt-get build-dep package：安装编译该包所需的编译依赖环境。
- sudo apt-get remove package：卸载软件。
- sudo apt-get autoremove：自动清理系统中不用的软件包。
- sudo apt-get autoclean：删除系统中旧版本软件包。
- apt-get --purge remove packagee：卸载软件，并删除配置文件。
- sudo apt-get upgrade：对所有软件进行自动升级。
- sudo apt-get dist-upgrade：对系统进行升级。
- sudo apt-get source package：获得软件的源代码。
- sudo apt-get clean：清除从源服务器下载的软件包，但是并不卸载已安装的文件。
- sudo apt-get check：更新缓冲区并检查损坏的软件包。

2．apt-cache 命令

apt-cache 是查询软件包缓存区信息的程序。该命令并不提供软件包安装、卸载之类的服务，但其提供了强大而丰富的查询功能，可以得到系统中软件包的各种信息。其具体用法如下：

```
apt-cache [选项] 命令
```

常用的“命令”如下所示。

- add：将一个软件包文件加入到软件包缓存区中。

- gencaches：生成软件包和源代码包的缓存信息。
- stats：显示统计信息，如软件包信息，软件包大小等。
- dump：显示所有缓存区中软件的内容。
- unmet：显示所有未满足的依赖关系。
- search：搜索指定的软件包，可以使用正则表达式。
- show ：简单介绍该软件包的相关信息。
- depends：显示软件包的依赖信息。
- rdepends：显示所有依赖于该包的软件包名字。
- pkgnames：显示所有软件包的名字。
- dotty：生成可用 GraphVis 处理的软件包关系图。
- xvcg：生成可用 xvcg 处理的软件包的关系图。
- policy：显示软件包的安装设置状态。

一些常用的方式如下所示（查询并不需要 root 权限）。

- apt-cache show package：显示软件包的相关信息，如说明、大小、版本等。
- apt-cache show -a package：显示软件的描述信息。
- apt-cache showsrc package：显示软件源码包的信息。
- apt-cache showpkg package：显示软件包的全部信息。
- apt-cache rdepends package：显示该软件被那些软件依赖。
- apt-cache dumpavail：可用软件包列表。
- apt-cache depends package：显示软件包的依赖情况。

3．aptitude

aptitude 也是一个强大的软件包管理软件，可以完成软件的安装，卸载和查询等工作。其具体用法如下：

```
aptitude [选项] <动作> ...
```

可用的选项有：

- -d：仅下载软件包，并不进行安装。
- -P：在执行任何动作前先确认。
- -y：在提示选择 yes/no 时，总是选择 yes。
- -f：试图修复依赖关系等损坏的软件包。
- -V：显示欲安装软件包的版本信息。
- -D：显示自动改变的软件包的依赖关系
- -Z：显示每个软件包的安装尺寸的变化。
- -u：运行程序时就下载新的软件包列表。
- -I：开始运行时执行安装。

aptitude 命令中可以指定的“动作”如下所示。

- install：安装软件包。这是 aptitude 命令中最为主要的一个动作，该动作不只能完成安装软件的功能，还能完成软件删除等工作。该参数有如下几种用法。

package+：安装软件包。.

<package>-：删除软件包。

<package>_：完全删除软件包，并且删除该软件的配置文件。

<package>=：锁定软件包，不让其进行任何的操作，如升级、删除等。并阻止软件包在未来的自动升级。

<package>:：保持当前软件包不变。与“锁定”不同的是，其并不阻止软件包在未来的自动升级。

<package>&M：标记软件包为自动安装。

<package>&m：标记软件包为手工安装。

- remove：卸载软件包。
- purge ：完全卸载软件包，包括删除软件包及其配置文件。
- markauto：将软件包标记为自动安装。
- unmarkauto：将软件包标记为手动安装。
- forbid-version：禁止某软件升级到某一特定版本。
- update：刷新软件缓冲区的软件包信息。
- safe-upgrade：安全升级。
- full-upgrade：完全升级。
- search：搜索软件包，可以使用简单的正则表达式。
- show：显示软件包的详细信息。
- clean：删除已下载的软件包文件。
- autoclean：删除旧版本的已下载软件包文件。
- changelog：查看一个软件包的更新记录。
- download：仅下载软件包的.deb 文件。
- reinstall：重新安装一个软件。

动作（如果未指定，aptitude 将进入交互模式），如图 8.18 所示。

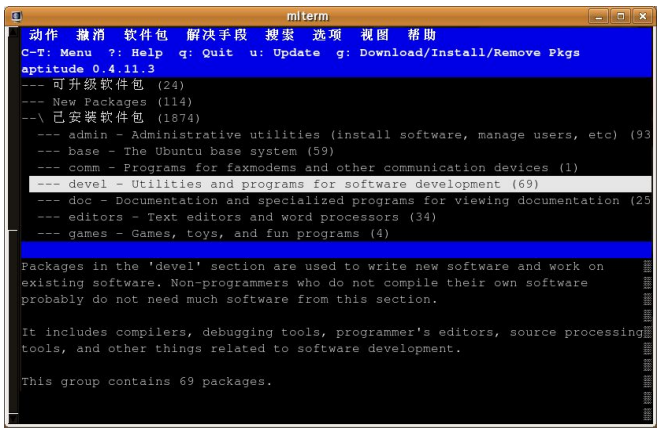


图 8.18　aptitude 界面

可以在如图 8.18 所示界面中按下 Ctrl+T 键打开菜单进行操作。其实这样的操作方式和“新立得”软件管理器很是类似的。只不过要用键盘移动光标位置而已。

8.5.2 dpkg 系列

dpkg 是一个庞大的家族，不仅提供了大量的参数选项，同时也提供了许多子命令如 dpkg-divert、dpkg-query、dpkg-split、dpkg-statoverride、start-stop-daemon 等来实现不同的功能。

1．安装 Deb 包

一般来说，dpkg 命令最常用的功能是安装 Deb 包。在终端中进入存放该 Deb 包的目录，然后运行“sudo dpkg -i 包名.deb”就可以完成安装 Deb 包的工作。

2．查询软件缓冲区中特定名称的软件包

以下两条命令的输出内容是完全相同的，包括了软件安装的目录及一些配置文件的位置等。

```
dpkg --listfiles package
dpkg-query -L package
```

部分输出如下所示。

```
earthfans@12:14 ～$ sudo dpkg --listfiles wicd
[sudo] password for earthfans:
/.
/etc
/etc/xdg
/etc/xdg/autostart
…
/usr/lib/pm-utils/sleep.d
/usr/lib/pm-utils/sleep.d/55wicd
```

输出的内容显示了包 wicd 相关的一些目录，包括/etc、//usr/lib/pm-utils/sleep.d 等，这些目录一般包含了该包的配置文件等内容。

3．查看软件包的详细信息

下列命令的输出包括了软件的安装状态、优先级、大小等信息。

```
dpkg -s package
dpkg-query -s package
```

部分输出如下所示。

```
earthfans@12:14 ～$ dpkg -s wicd
Package: wicd                                          #包名称
Status: install ok installed                           #状态是已经安装成功
Priority: extra                                        #优先级
Section: net                                           #分类
Installed-Size: 1804                                   #软件大小
Maintainer: Adam Blackburn <compwiz18@gmail.com>       #开发者
Architecture: all                                      #架构
Version: 1.5.8                                         #版本
Depends: python, python-gtk2, python-dbus | python2.4-dbus, wpasupplicant,
python-glade2, wireless-tools                          #依赖关系
Conflicts: gtkwifi, network-manager, wifi-radar        #关键词
```

```
 Conffiles:
  /etc/xdg/autostart/wicd-tray.desktop 44c8fb33744797e4db1d0e85b20ce66a
 …
  /etc/wicd/encryption/templates/peap-tkip d28f1b19d0c33b9768e28af1beb95bad
 Description: wired and wireless network manager        #描述信息
  Wicd supports wired and wireless networks, and capable of creating and
  tracking profiles for both.  It has a template-based wireless encryption
  system, which allows the user to easily add encryption methods used.
  It ships with some common encryption types, such as WPA 1 and 2 and WEP.
  Wicd will automatically connect at startup to any preferred network within
  range.
```

4．查看系统中软件包状态

下列命令显示了软件的状态，包括版本及描述信息等：

```
dpkg -l package
dpkg-query -l package
```

部分输出如下所示：

```
earthfans@12:16 ~$ dpkg -l wicd
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Cfg-files/Unpacked/Failed-cfg/Half-inst/trig-aWait/Trig
-pend
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase
=bad)
||/ Name          Version       Description
+++-==============-==============-============================================
=======
ii  wicd          1.5.8         wired and wireless network manager
```

上述输出显示了 wicd 的版本号是 1.5.8，而且其是无线及有线的网络管理器。

5．查看某个文件的归属包

下列命令可以显示出属于包的文件的内容。

```
dpkg-query -S package
dpkg -S package
```

部分输出如下所示。

```
earthfans@12:23 ~$ dpkg -S wicd
wicd: /usr/lib/python2.5/site-packages/wicd/gui.py
wicd: /usr/share/locale/pl/LC_MESSAGES/wicd.mo
wicd: /usr/share/locale/lt/LC_MESSAGES/wicd.mo
…
wicd: /usr/share/locale/eu_ES/LC_MESSAGES/wicd.mo
wicd: /usr/share/pixmaps/wicd/idle-bad-signal-lock.png
wicd: /usr/share/pixmaps/wicd/receiving-bad-signal-lock.png
```

上述输出显示了系统中哪些文件是安装 wicd 时安装的，如 gui.py、wicd.mo 等。

6．软件卸载与清除

Ubuntu 系统中软件卸载与清除是两个不同的概念，虽然两者功能均是卸载软件包。但清除软件包同时进行软件卸载以及删除软件配置文件，而卸载软件包并不删除软件配置文件。

卸载软件包的命令如下：

```
dpkg -r package
```

清除软件的命令如下：

```
dpkg -P package。
```

8.6 小结

本章介绍了 Linux 系统进行软件安装与管理的相关知识，包括源服务器的配置，以及丰富的软件安装与卸载的方法。本章的重点内容是新立得软件管理器，这也是 Linux 系统中进行软件管理的最为主要的工具。掌握该软件的使用可以给 Linux 的使用带来极大的便利，因为掌握该软件后就可在数以万计的软件库中轻易找到适合自己的软件。本章的难点是“编译、安装提供源码安装的软件”。因为提供源码的软件可能在编译时要满足一定的条件，而满足这些条件有时是件很困难的工作，这就要求用户要对软件的依赖关系有深刻的了解，这些问题的解决有时只能依靠用户积累的经验。

第 9 章　磁 盘 管 理

磁盘是计算机系统中数据存储的一种设备。磁盘的速度比光盘快，价格又比内存便宜，具有很高的性价比。磁盘存放的数据在断电后不消失，可以用来安装操作系统等系统软件。由于上述优点，磁盘在现代计算机系统中得到了越来越多的应用，成为计算机系统中最主要的存储设备。本章将介绍对磁盘进行管理的方法，如分区、格式化、分析磁盘使用情况，以及优化磁盘性能的一些基本方法。

9.1　用 Live CD 进行磁盘管理

Ubuntu 的光盘一般均具有 Live CD 功能，也就是可以不用实际安装，就可以从光盘中启动和运行一个完整的 Linux 系统。这种功能一方面提供了试用 Ubuntu 的机会，另一方面也为系统维护提供了一个强大的平台。本节介绍使用 Live CD 启动后进行磁盘管理的方法。

9.1.1　磁盘分区

新出厂磁盘对于操作系统来说，是不可直接存取数据的。因为磁盘里面没有操作系统所需要的各种数据结构，如分区、格式等相关信息。所以在使用一块新磁盘前要对其进行分区、格式化等操作。

如果要对一块磁盘进行分区，可以将其挂载到另一个系统，然后使用相应的分区软件进行分区。如果要对一台没有安装操作系统的裸机的磁盘进行分区，最为简便实用的方法是用 Live CD 启动 Linux 系统，然后在该系统中进行相应的分区操作。

将 Ubuntu 的 Live CD 光盘放入计算机，然后打开电源启动系统，选择使用 Live CD 启动计算机，就会进入与普通的 Ubuntu 看起来没有差别的光盘版 Ubuntu。依次选择主菜单的“System”|“Administration”|“Partition editor”命令，就可以启动 Gparted 分区编辑器进行分区管理。该软件的主界面如图 9.1 所示。

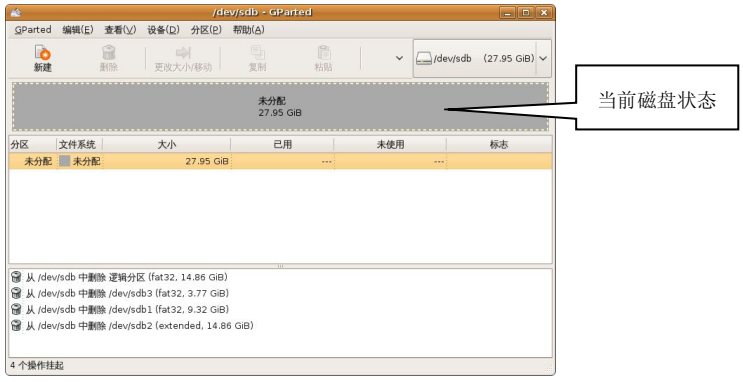


图 9.1　Gparted 分区管理器

图 9.1 显示了一个空白的磁盘，其容量是 27.95GB。“未分配”表明这是一个尚未建立分区信息的磁盘。下面就以该磁盘为例介绍磁盘的分区等操作。



一块磁盘最多可以建立 4 个主分区。主分区对于 Ubuntu 来说并不是必需的，但 Windows 与 freeBSD 等必须安装在主分区中，因此如果要安装这些操作系统，就需要划分相应的主分区。

在分区前，应该先对磁盘进行相应的规划。一般情况下，Windows XP 或 Ubuntu 操作系统所在的系统分区应该分配 10GB 左右的空间，最好再分配一个专门用于存储数据的分区，再有一个用于存储影音资料类的分区。分区规划好后，就可开始分区工作了。

单击图 9.1“当前磁盘状态”区域，使其成为当前磁盘，然后单击工具栏上 New（新建分区）按钮，开始建立该磁盘上的第一个分区。此时，会弹出图 9.2 所示“新建分区”对话框。



图 9.2　新建分区

对于用户来说，图 9.2 中重要的选项有以下几个。

- 新大小（MB）：输入新建分区的大小，以 MB 为单位。
- 创建为：分区的类型，用户可以选择建立一个主分区，或者选择建立扩展分区。
- 文件系统：这是用来选择分区的格式，Windows 常用的格式有 fat32、NTFS，而 Linux 常用的格式有 ext3 等。
- 标签：分区卷标。
- 舍入到柱面：在划定分区时，可以勾选此项以自动避开柱面边界，也就是说会调整分区的大小，以使一个柱面只位于一个分区。选择此项，可以部分地提高性能。

本例中，设定分区大小为 10GB（10000MB），分区类型为主分区（这是系统的第一个分区，一般应将其设置为主分区），分区格式选择了 ext2。这时，应该可以看到磁盘状态显示区域发生了变化，出现的白色区域代表用户新建的分区，而灰色区域代表还未进行分配的区域。设置好选项后，单击“添加”按钮，结束该分区的创建。

继续单击图 9.1 中“当前磁盘状态”区域中代表还未进行分区的灰色区域，使之成为当前操作区域。然后继续单击工具栏上的“新建”按钮建立其他的分区。对该磁盘的一个可能的分区结果如图 9.3 所示。

9.1.2　格式化

完成了磁盘的分区操作，接下来就应该对各分区进行格式化操作。完成格式化操作后，磁盘才会真正被各操作系统所识别。在 Gparted 软件中进行格式化操作也是很容易的工作。

下面以一个实际使用的磁盘来介绍格式化分区的方法。该磁盘的格式化分区情况如图 9.4 所示。假设本次操作要格式化的是 sda9。格式化的具体步骤如下所示。

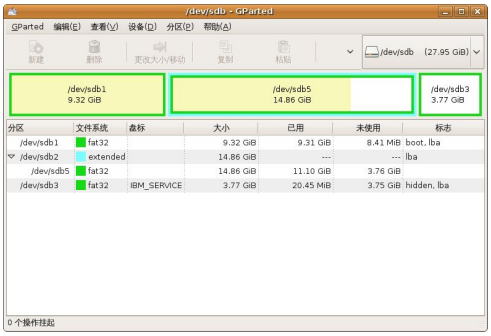


图 9.3　磁盘分区示意图

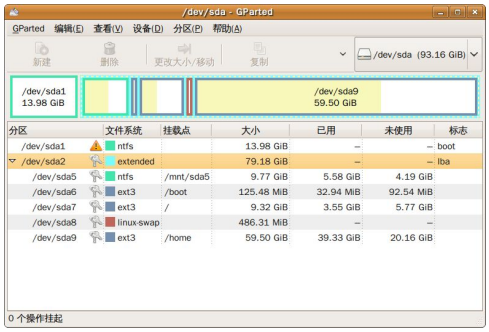


图 9.4　格式化操作示例

（1）右击/dev/sda9 项，在弹出的快捷菜单中选择 Umount 命令就可卸载该分区。



/dev/sda9 正在使用，所以其分区标志前有一个钥匙标志。有这个标记的分区是无法进行任何系统级的操作（如格式化、更改大小）。所以，应该先将其卸载。卸载分区后，分区前的钥匙标志消失了，然后就可对该分区执行系统级的操作。

（2）右击/dev/sda9 项，就可以发现刚才还不可用的 Format to 菜单项已经可用，将鼠标停留在 Format to 菜单项 1 秒后，会弹出选择分区格式的快捷菜单。如选择 NTFS 命令，就可将该分区格式化为 NTFS 格式。

（3）选择菜单“Edit”|“Apply all operation”命令，将刚才所设置的更改实际写入到磁盘中。因为前述操作完成后，磁盘还没有发生任何变化，所以需要执行该命令使前述设置生效。

9.1.3　更改分区大小

如果对于分区的大小不太满意，可使用 Gparted 对其进行更改。要注意的是如果要增大分区的尺寸，则硬盘中必须要有空闲空间可供使用。而如果进行的是缩小尺寸的操作，如将 30GB 的分区减小为 18GB，则其余的 12GB 会成为空闲空间。分区大小调整的操作过程如下所示。

（1）先检查其是否已经被挂载，如果被挂载，则参照格式化操作中相关内容卸载分区。

（2）在分区上右击，在弹出的快捷菜单中选择 Resize/Move 命令，会弹出与图 9.1 类似的对话框。

（3）在 New Size 文本框中输入该分区的新的大小。

（4）单击 Add 按钮即可完成更改分区大小的工作。

（5）将更改写入磁盘的方法可参照前述内容进行操作。

一般来说，更改分区大小要比分区或格式化花费更多的时间。



磁盘安装 Ubuntu 后，会发现默认并未安装 Gparted，需要运行 sudo apt-get install gparted 命令来进行安装，然后才可以在主菜单中找到该程序。

9.2　磁盘管理命令

9.1 节介绍了使用 Ubuntu 的 Live CD 启动，然后对磁盘进行分区等操作的过程。除了用图形界面的软件进行磁盘的操作外，Ubuntu 还提供了一些用于磁盘操作的命令，提供了更为高效的磁盘管理功能。本节以分区命令 fdisk 与格式化命令 mkfs 为例进行简单的介绍。

9.2.1 分区命令

磁盘的分区操作是一个比较复杂的过程，因简单用文字来叙述是一件很困难的事，所以下面用一些使用 fdisk 进行分区操作的实际例子来理解分区的过程。

1．删除原有的分区

有时为了重新划分分区，需要将已经存在的分区删除。Linux 提供了 fdisk 命令用于分区操作，当然其中包含了删除分区的功能。下面的示例是使用 fdisk 删除分区的步骤（以#开头的是注释）。

（1）使用 root 权限，运行 fdisk 命令：

```
earthfans@mylaptop:~$ sudo fdisk /dev/sda
```

fdisk 需要 root 权限，在命令后要跟操作目标磁盘，本例中是/dev/sda。

（2）由于使用了 root 权限，所以可能需要输入密码，在提示后输入用户登录密码即可。

```
[sudo] password for earthfans:                    #输入密码
```

（3）fdisk 程序会显示硬盘基本情况，并提示用户输入相应的命令。输出的内容如下所示：

```
The number of cylinders for this disk is set to 12161.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)
```

（4）在 fdisk 提示输入操作命令时，输入 m，然后回车显示帮助菜单。显示的帮助菜单如下所示：

```
Command (m for help): m                  #提示输入命令,输入 m 会显示帮助菜单,
Command      action
a            toggle a bootable flag                  #设置当前分区为可启动分区
b            edit bsd disklabel                      #编辑分区卷标
c            toggle the dos compatibility flag       #设置为 DOS 兼容模式
d            delete a partition                      #删除分区
l            list known partition types              #显示分区类型
m            print this menu                         #显示本菜单
n            add a new partition                     #添加新分区
o            create a new empty DOS partition table  #建立空的 DOS 分区表
p            print the partition table               #显示分区表
q            quit without saving changes             #退出而且不应用更改
s            create a new empty Sun disklabel        #建立空的 Sun 兼容的卷标
t            change a partition's system id          #更改分区的系统 ID,也就是
                                                      更换分区类型
u            change display/entry units              #改变显示单位
v            verify the partition table              #验证分区表
w            write table to disk and exit            #将卷标写入磁盘,并退出
x            extra functionality (experts only)      #高级模式
```

（5）在分区前先查看当前分区的情况，在 fdisk 命令提示符后输入 p，如下所示：

```
Command (m for help): p
```

fdisk 会显示当前磁盘的分区情况，如下所示：

```
Disk /dev/sda: 100.0 GB, 100030242816 bytes     #磁盘大小为 100GB
255 heads, 63 sectors/track, 12161 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0xf0b1ebb0                     #磁盘识别码为 0xf0b1ebb0
Device BootStartEndBlocksId  System
/dev/sda1*  1  1825 14659281 7  HPFS/NTFS
/dev/sda21826 12161 83023920 f  W95 Ext'd (LBA)
/dev/sda51826  3100 10241406 7  HPFS/NTFS
/dev/sda63101  3116128488+  83  Linux
/dev/sda73117  4332  9767488+  83  Linux
/dev/sda84333  4394497983+  82  Linux swap / Solaris
/dev/sda94395 12161 6238839683  Linux
```

（6）删除旧的分区，方法是在 fdisk 命令提示符后输入 d，如下所示：

```
Command (m for help): d
```

（7）fdisk 会提示输入要删除的分区的编号，输入要删除分区的编号即可：

```
Partition number (1-9): 5
```

输入要删除的分区的编号，就是磁盘符“sda”后面的数字，输入 5 意思是删除/dev/sda5。

（8）在 fdisk 命令提示符中输入命令 w，将刚才所做的更改写入磁盘：

```
Command (m for help): w
```

fdisk 会将更改写入磁盘，然后退出。

2．在新磁盘上建立分区

前面介绍的是在已经分好区的磁盘上操作的实例，但很多时候，我们需要对一个新磁盘进行分区的操作。所以，再来介绍使用 fdisk 操作一个新磁盘的工作过程。

（1）使用 root 身份运行 fdisk，操作目标是 sdb1，命令如下：

```
westnorth@09:07 ~$ sudo fdisk /dev/sdb                  #打开 sdb 进行操作
 [sudo] password for westnorth:                         #输入密码
#下面的内容是 sdb 的信息
The number of cylinders for this disk is set to 3649.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)
```

（2）在分区前应该先检查一下磁盘当前的分区情况。fdisk 的子命令 p 可以实现此功能。在 fdisk 的命令提示符中输入 p 即可：

```
Command (m for help): p                                    #显示当前分区情况
Disk /dev/sdb: 30.0 GB, 30020272128 bytes
255 heads, 63 sectors/track, 3649 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0xffffffff
Device BootStartEndBlocksId  System
```

从输出中可以看出，这是一个 30GB 的新磁盘，还没有进行分区的划分。

（3）fdisk 的子命令 n 是用来创建新分区的命令，在 fdisk 命令提示符中输入 n 开始新建磁盘分区。

```
Command (m for help): n
#n 是新建分区指令,新建分区包括建立主分区、扩展分区等内容
```

（4）输入 n 后，fdisk 会显示如下内容，提示用户选择分区类型。e 为扩展分区，p 为主分区。在提示符后输入 p 开始建立主分区。过程如下所示：

```
Command action
e   extended
p   primary partition (1-4)
p                                                          #p 的意思是新建一个主分区
```

（5）fdisk 会提示输入主分区编号，输入 1 即可。

```
Partition number (1-4): 1                                  #主分区的编号,输入数字 1
```

（6）fdisk 提示输入起始柱面，由于新建的是第一个主分区，所以选择默认值 1 即可。

```
First cylinder (1-3649, default 1):          #提示输入起始柱面,直接回车取默认值
Using default value 1
```

（7）fdisk 提示输入分区的结束柱面，这个结束柱面的数字决定了磁盘的大小。不过，一般用户对柱面的概念比较模糊。所以 fdisk 也允许用户输入“+磁盘大小”格式的内容来决定分区大小，如下所示：

```
Last cylinder or +size or +size M or +size K (1-3649, default 3649): +10000M
#提示输入分区的结束柱面,可以输入+10000M 的形式,意思是从起始柱面起 10G。
```

（8）成功创建主分区后，fdisk 回到命令提示符状态。此时应该输入 n 继续创建分区。而本次新建的磁盘分区应为扩展分区。

```
Command (m for help): n                                    #n 的意思是继续创建分区
```

（9）fdisk 显示如下内容，意义如前所述。而本次应该输入 e，意思是创建扩展分区。

```
Command action
e   extended                                               #e 是建立扩展分区
p   primary partition (1-4)                                #p 是建立主分区
e                                                          #输入命令 e 意思是创建扩展分区
```

（10）fdisk 提示输入分区编号，一般扩展分区应该输入 2。同时也会要求用户输入开始及结束柱面信息等，由于与前述内容相同，这里就不将这些步骤分开介绍了。具体输入如下所示：

```
Partition number (1-4): 2                          #扩展分区编号,本次输入数字 2
First cylinder (1218-3649, default 1218):
#扩展分区的起始柱面紧跟着主分区的结束柱面,仍然按回车键取默认值即可
Using default value 1218
Last cylinder or +size or +sizeM or +sizeK (1218-3649, default 3649):
#输入扩展分区的结束柱面,一般来说,扩展分区应该使用剩余的所有空间,所以本次仍然选择默认值
Using default value 3649
```

（11）现在创建了磁盘的主分区与扩展分区。但操作系统可以直接操作的应该是逻辑分区。而逻辑分区应该在扩展分区中创建。现在，应该创建逻辑分区。同样，在 fdisk 中输入 n 开始创建新分区，本例的目标是创建逻辑分区。

```
Command (m for help): n                            #再创建分区,本次是创建逻辑分区
```

（12）fdisk 显示如下信息。l 是创建逻辑分区，而 p 是创建主分区。本次输入 l，意思是创建逻辑分区。fdisk 的显示及输入如下所示：

```
Command action
l   logical (5 or over)
p   primary partition (1-4)
l                                                  #选择 l 意思是创建逻辑分区
```

（13）接下来要设置的项目是逻辑分区的大小。与前面介绍的内容相当类似，均是输入起始柱面数即可，如下所示：

```
First cylinder (1218-3649, default 1218):
#设置扩展分区的起始、结束柱面。一般来说,逻辑分区应该从扩展分区的起始柱面开始创建
Using default value 1218
Last cylinder or +size or +sizeM or +sizeK (1218-3649, default 3649):#直接按回车键,取默认值
Using default value 3649
```

本次使用了所有的扩展分区来创建了一个逻辑分区，也可以参照上面的操作输入+xxxxM 来创建相应大小的逻辑分区。

（14）现在应该显示刚刚建立的分区情况，检查分区是否已经按照前述步骤正确建立。参照上述内容，fdisk 中显示分区状况使用的命令是 p。输出情况如下：

```
Command (m for help): p                            #输入 p,显示磁盘分区情况
Disk /dev/sdb: 30.0 GB, 30020272128 bytes
255 heads, 63 sectors/track, 3649 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0xffffffff
Device  Boot        Start   EndBlocks           Id  System
/dev/sdb1   1       1217    9775521             83  Linux
/dev/sdb2           1218    3649    19535040    5  Extended
/dev/sdb5           1218    364919535008+       83  Linux
```

由如上输出可以看到，已经创建了主分区与相应的逻辑分区。

（15）现在新建工作已经完成，但实际上所有的操作均未写入磁盘。还需要使用 w 命令使

fdisk 将所做更改写入磁盘。这一步，可能看起来有点多余，其实，这是非常必要的。因为有时用户可能在执行过程中进行了误操作，通过 p 命令查看如果有错误，还可以取消操作来避免更大的损失。如果各种操作均正确无误，就可以正式将所做更改写入磁盘。这样，可以在最大限度上保证操作的安全性。具体的操作及输出如下所示：

```
Command (m for help): w
#输入 w,将所做的更改写入磁盘,相对来说,这个过程要比 gparted 要快许多
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

3．更改分区格式

fdisk 也可进行更改分区格式的工作，其实更改分区格式就是更改分区的系统编号。其步骤如下所示。

（1）进入 fdisk，fdisk 会显示命令提示符。在 fdisk 命令提示符中输入 t 开始更改分区格式。

```
Command (m for help): t                          #输入 t 意思是更改分区格式
```

（2）fdisk 会提示用户输入要更改格式的分区编号，本例输入 1，意思是更改主分区（分区编号为 1）的格式。

```
Partition number (1-5): 1                        #输入 1 意思是更改主分区的格式
```

（3）fdisk 的分区格式是用数字来表示的。如 NTFS 的编号是 7、FAT32 的编号是 c 等。要更改分区格式，只要将分区的编号更改即可。由于编号比较难于记忆，所以可以先可用 l 命令来查看 fdisk 更改的所有分区格式。

```
Hex code (type L to list codes): l               #输入字母 l 显示可用格式列表
 0  Empty  1e  Hidden W95 FAT1 80  Old Minix be  Solaris boot
 1  FAT12  24  NEC DOS81  Minix / old Lin bf  Solaris
 2  XENIX root39  Plan 9 82  Linux swap / So c1  DRDOS/sec (FAT-
 3  XENIX usr 3c  PartitionMagic  83  Linux  c4  DRDOS/sec (FAT-
 4  FAT16 <32M40  Venix 80286  84  OS/2 hidden C:  c6  DRDOS/sec (FAT-
 5  Extended  41  PPC PReP Boot85  Linux extended  c7  Syrinx
 6  FAT16  42  SFS 86  NTFS volume set da  Non-FS data
 7  HPFS/NTFS 4d  QNX4.x 87  NTFS volume set db  CP/M / CTOS / .
 8  AIX 4e  QNX4.x 2nd part 88  Linux plaintext de  Dell Utility
 9  AIX bootable 4f  QNX4.x 3rd part 8e  Linux LVM df  BootIt
 a  OS/2 Boot Manag 50  OnTrack DM93  Amoeba e1  DOS access
 b  W95 FAT32 51  OnTrack DM6 Aux 94  Amoeba BBTe3  DOS R/O
 c  W95 FAT32 (LBA) 52  CP/M9f  BSD/OS e4  SpeedStor
 e  W95 FAT16 (LBA) 53  OnTrack DM6 Aux a0  IBM Thinkpad hi eb  BeOS fs
 f  W95 Ext'd (LBA) 54  OnTrackDM6a5  FreeBSDee  EFI GPT
10  OPUS55  EZ-Drive  a6  OpenBSDef  EFI (FAT-12/16/
11  Hidden FAT12 56  Golden Bowa7  NeXTSTEP  f0  Linux/PA-RISC b
12  Compaq diagnost 5c  Priam Edisk  a8  Darwin UFSf1  SpeedStor
14  Hidden FAT16 <3 61  SpeedStor a9  NetBSD f4  SpeedStor
16  Hidden FAT16 63  GNU HURD or Sys ab  Darwin boot  f2  DOS secondary
17  Hidden HPFS/NTF 64  Novell Netware  b7  BSDI fsfd  Linux raid auto
```

```
18  AST SmartSleep  65  Novell Netware  b8  BSDI swap fe  LANstep
1b  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid ff  BBT
1c  Hidden W95 FAT3 75  PC/IX
```

（4）得到 NTFS 的编号为 7。输入 7 就将选中分区（本例是 1，是第 2 步时设置的）更改为 NTFS 格式。

```
Hex code (type L to list codes): 7                    #输入 7,将分区更改为 NTFS
Changed system type of partition 1 to 7 (HPFS/NTFS)#显示确认信息
```

（5）使用命令 p，显示当前磁盘的分区信息，查看是否已经将该分区更改为 NTFS。

```
Command (m for help): p                               #显示分区
Disk /dev/sdb: 30.0 GB, 30020272128 bytes
255 heads, 63 sectors/track, 3649 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0xffffffff
Device  Boot        Start   End             BlocksId    System
/dev/sdb1   1       1217    9775521             7       HPFS/NTFS   # 已 经
将 sdb1 更改为 NTFS 格式
/dev/sdb2           1218    3649 19535040       5       Extended
/dev/sdb5           1218    3649 19535008       +83     Linux
```

（6）使用 w 命令向磁盘写入信息。一般来说，所有对于磁盘的操作均要执行 w 命令后才会正式写入磁盘。操作及输出如下所示：

```
Command (m for help): w
#输入 w,将所做的更改写入磁盘,相对来说,这个过程要比 gparted 要快许多
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

9.2.2　格式化命令

fdisk 只是一个建立分区的命令，其并未提供对分区进行格式化操作的功能。要对分区进行格式化，就要使用 mkfs 命令。mkfs 是一个建立 Linux 文件系统的命令，可以将分区格式化成 ext3 或 reiserfs 等格式。将分区 sdb1 格式化为 ext3 的示例如下所示。

```
westnorth@10:00 ～$ sudo mkfs -t ext3 /dev/sdb1          #将第二块磁盘的第 1
个分区#格式化为 ext3 格式
#以下是命令的输出信息,显示了命令的执行过程
mke2fs 1.40.8 (13-Mar-2008)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)                          #块大小为 4096
Fragment size=4096 (log=2)                       #碎片尺寸为 4096
612000 inodes, 2443880 blocks                    #有 612000 个 i 节点,2443880 个块
122194 blocks (5.00%) reserved for the super user  #122194 个块保留给超级用户
First data block=0
```

```
    Maximum filesystem blocks=2503999488
    75 block groups
    32768 blocks per group, 32768 fragments per group
    8160 inodes per group
    Superblock backups stored on blocks:        #超级块备份在下列块中
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
    Writing inode tables: done                  #写入 i 节点
    Creating journal (32768 blocks): done       #创建日志
    Writing superblocks and filesystem accounting information:     #写入超级块
    done
    This filesystem will be automatically checked every 37 mounts or 180 days,
whichever comes first.  Use tune2fs -c or -i to override.
    #该分区每挂载 37 次或使用 180 天后检查一次,取决于哪一个条件先达到。可以使用 tue2fs -c
或-i 来覆盖此选项。
```

用 mkfs 还可以将磁盘格式化成 reiserfs 的格式。使用如下命令可将分区 sdb5 格式化为 Reiserfs 格式。

```
    sudo mkfs -t reiserfs /dev/sdb5
```

该命令的输出与格式化为 ext3 的输出基本类似，就不再重复说明了。有兴趣的读者可能参见格式化为 ext3 示例的注释。



mkfs 是还拥有一系列的子命令，可以直接完成相应格式的格式化功能。这些命令包括 mkdosfs、mkfs.ext2、mkfs.ext3、mkfs.minix、mkfs.msdos、mkfs.vfat、mkfs.xfs、mkfs.xiafs 等，如 mkfs.ext3 可直接将分区格式化为 ext3 格式。

9.3 查看磁盘使用情况

磁盘管理，除了对磁盘进行分区与格式化操作外，还有一个重要的内容是查看磁盘的使用情况。只有清楚地知道磁盘的使用情况，才能有效地对磁盘进行管理。本节介绍一些用来查看磁盘基本情况的命令，如查看磁盘分区情况、磁盘使用情况，以及查看磁盘的性能参数等命令。

9.3.1 fdisk 查看分区情况

在 9.2.1 节中介绍过 fdisk 可以用来进行分区操作，如删除、新建分区等。但其实 fdisk 也可用来查看分区的详细情况，而且其使用简单、输出信息简明易读，是普通用户首选的查看分区的命令。使用 fdisk 查看分区需要 root 权限，可以使用 sudo 来切换，显示分区情况的参数是 l。使用 fdisk 查看分区的具体命令及输出如下所示：

```
    earthfans@mylaptop:~$ sudo fdisk -l         #显示分区情况
    [sudo] password for earthfans:
    #下面是一些磁盘的技术参数,如大小、柱面数、磁盘识别码等信息
    Disk /dev/sda: 100.0 GB, 100030242816 bytes
    255 heads, 63 sectors/track, 12161 cylinders
    Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Disk identifier: 0xf0b1ebb0
#下面输出的意义,可以参见输出结果后的说明
Device  Boot Start End Blocks Id  System
/dev/sda1 *  1  1825 14659281 7  HPFS/NTFS
/dev/sda21826 12161 83023920 f  W95 Ext'd (LBA)
/dev/sda51826  3100 10241406 7  HPFS/NTFS
/dev/sda63101  3116128488+  83  Linux
/dev/sda73117  4332  9767488+  83  Linux
/dev/sda84333  4394497983+  82  Linux swap / Solaris
/dev/sda94395 12161 6238839683  Linux
```

其实 fdisk 的输出是比较直观的，不过为了方便理解，还是简单地介绍一下部分输出项的意义。

- Device：设备，也就是分区。
- Boot：分区是否是启动分区。上例中/dev/sda1 是可启动分区。每个磁盘中只有一个分区应该是可启动分区。
- Id：分区格式编号。如 7 代表 NTFS，83 代表 Linux 分区。
- System：分区格式。

9.3.2　查看/dev/disk 目录相关内容

Ubuntu 系统的/dev/disk 目录以不同的方式显示着系统中磁盘的分区情况。该目录中有 4 个子目录，分别有不同的意义，如 by-label 目录中的文件是以分区卷标形式显示各分区等。下面分别进行介绍。

1．查看以卷标形式显示的分区情况

卷标也就是磁盘分区的名称。有时为了管理的方便，会给磁盘分区命名，这就是磁盘的卷标。by-label 子目录可以以卷标为依据显示相关的磁盘信息，查看方法如下：

```
~ $ ls -l /dev/disk/by-label/
total 0
lrwxrwxrwx1 root  root  10 12 月  5 02:16 software -> ../../sda5
```

从输出可以看到，该目录中有一个名为 software 的链接文件。software 实际上是 sda5 的卷标，而该文件实际是/dev/sda5 分区的一个符号链接。

2．查看以硬件信息形式显示的分区情况

by-path 子目录能以 pci 设备编号的形式显示磁盘的分区情况。详情如下：

```
~ $ ls -l /dev/disk/by-path/
total 0
lrwxrwxrwx1 root  root  10 12 月  5 02:16 pci-0000:00:1f.1-scsi-0:0:0:0
-> ../../scd0
lrwxrwxrwx1 root  root  9 12 月  5 02:16 pci-0000:00:1f.2-scsi-0:0:0:0
-> ../../sda
lrwxrwxrwx1 root  root  10 12 月  5 02:16 pci-0000:00:1f.2-scsi-0:0:0:0-
part1 -> ../../sda1
```

```
    lrwxrwxrwx1 root  root  10 12 月  5 02:16 pci-0000:00:1f.2-scsi-0:0:0:0-
part2 -> ../../sda2
    lrwxrwxrwx1 root  root  10 12 月  5 02:16 pci-0000:00:1f.2-scsi-0:0:0:0-
part5 -> ../../sda5
    lrwxrwxrwx1 root  root  10 12 月  5 02:16 pci-0000:00:1f.2-scsi-0:0:0:0-
part6 -> ../../sda6
    lrwxrwxrwx1 root  root  10 12 月  5 02:16 pci-0000:00:1f.2-scsi-0:0:0:0-
part7 -> ../../sda7
    lrwxrwxrwx1 root  root  10 12 月  5 02:16 pci-0000:00:1f.2-scsi-0:0:0:0-
part8 -> ../../sda8
    lrwxrwxrwx1 root  root  10 12 月  5 02:16 pci-0000:00:1f.2-scsi-0:0:0:0-
part9 -> ../../sda9
```

由输出可知，该目录中也包含着一些链接文件，链接到系统的一些块设备，其实也就是链接到相应的分区。

3．查看以 UUID 形式显示的分区情况

UUID 是一个磁盘分区的识别信息码，这个号码是唯一的、不可能重复的，所以被广泛用在需标识硬件信息的场合。by-uuid 子目录是以 UUID 形式显示磁盘分区信息，可用如下方式查看。

```
    ～ $ ls -l /dev/disk/by-uuid/
    total 0
    lrwxrwxrwx1 root   root   10 12 月   5 02:16 1a63a3c2-bd31-45aa-9451-
b86123d7c8918 -> ../../sda9
    lrwxrwxrwx1 root   root   10 12 月   5 02:16 8b1233b8-b217-43a1f-9561
-e0612347ef9a3 -> ../../sda8
```

由输出可知，该目录中是一些文件，以磁盘的 UUID 为名称，连接到相应的分区。如果需要分区的 UUID 信息，可以查看该文件来获得。如 sda8 的 UUID 是 8b1233b8-b217-43a1f-9561-e0612347ef9a3。

4．查看以磁盘 id 形式显示的分区情况

磁盘 id 同样也是磁盘的身份标识，而且其还显示了磁盘品牌等信息。可用如下方法以磁盘 id 形式显示分区情况。

```
    ～ $ ls -l /dev/disk/by-id/
    total 0
    lrwxrwxrwx1 root   root   9 12 月   5 02:16 ata-Hitachi_HTS541210H9SA00
_HP2D00BDG2EV5A -> ../../sda
    lrwxrwxrwx1 root   root   10 12 月   5 02:16 ata-Hitachi_HTS541210H9SA00
_HP2D00BDG2EV5A-part1 -> ../../sda1
    lrwxrwxrwx1 root   root   10 12 月   5 02:16 ata-Hitachi_HTS541210H9SA00
_HP2D00BDG2EV5A-part2 -> ../../sda2
    lrwxrwxrwx1 root   root   10 12 月   5 02:16 ata-Hitachi_HTS541210H9SA00
_HP2D00BDG2EV5A-part5 -> ../../sda5
    lrwxrwxrwx1 root   root   10 12 月   5 02:16 ata-Hitachi_HTS541210H9SA00
_HP2D00BDG2EV5A-part6 -> ../../sda6
```

```
    lrwxrwxrwx1 root   root   10 12 月   5 02:16 ata-Hitachi_HTS541210H9SA00
_HP2D00BDG2EV5A-part7 -> ../../sda7
    lrwxrwxrwx1 root  root  10 12 月  5 02:16
```

由输出可知，系统安装的磁盘是东芝公司出品、第一部分被划分成 sda1 等。

9.3.3　查看/proc/partitions 中的分区信息

在/proc 目录中，记录了许多系统重要的信息，其中也包括了磁盘的分区信息。可以通过查看/proc/partions 文件来获取。

```
    ~ $ cat /proc/partitions
    major   minor   #blocks     name
    8       0       97685784    sda
    8       1       14659281    sda1
    8       2       1           sda2
    8       5       10241406    sda5
    8       6       128488      sda6
    8       7       9767488     sda7
    8       8       497983      sda8
    8       9       62388396    sda9
```

由输出易知，系统中有一块磁盘，共分为 6 个区。其中，sda1 是主分区，sda2 是扩展分区，sda5-sda9 是逻辑分区。

9.3.4　查看磁盘已挂载分区使用情况的命令 df

df 命令也是用于查看磁盘分区情况的，但与 fdisk 不同的是，其只显示已挂载分区的信息；而且不只可以显示磁盘块信息，还可以显示 i 节点的相关信息。其用法如下：

```
    df  [选项]
```

df 命令可用的选项列表如下所示。

- -a：显示所有文件系统的磁盘使用情况，包括动态生成的目录，如/proc 文件系统等。
- -k：以 k 字节为单位显示。
- -i：显示 i 节点信息。
- -t：显示指定类型分区的磁盘使用情况。
- -x：显示不是指定类型分区的磁盘空间使用情况。
- -T：显示文件系统类型。

如下命令可以显示文件系统的类型，同时还显示该分区的使用情况，以及该分区的挂载点。

```
    /dev/disk $ df -T           #显示文件系统类型
    文件系统  类型         1K-块    已用      可用      已用%    挂载点
    /dev/sda7   ext3     9614116 4249108  4876636 47%         /
    tmpfs       tmpfs    1031664 4        1031660 1%       /lib/init/rw
    varrun      tmpfs    1031664 216      1031    448 1%   /var/run
    varlock     tmpfs    1031664 0        1031664 0%       /var/lock
    …
```

限于篇幅，输出只是截取了一部分，从这部分可以看出输出的格式。由以上输出可知，sda7 的文件类型是 ext3，tmpfs 的文件类型是 tmpfs，而且从输出中也易知分区的使用情况等信息。

如下命令可以输出当前文件系统的 i 节点信息。同样也输出了相关分区 i 节点的使用情况等信息。

```
/dev/disk $ df -ia          #显示全部文件系统的 i 节点信息
文件系统      Inode    (I)已用 (I)可用 (I)已用% 挂载点
/dev/sda7    610800   175893    434907     29%      /
tmpfs        223337        4    223333     1%       /lib/init/rw
/proc        0             0         0     -        /proc
sysfs        0             0         0     -        /sys
varrun       223337       69    223268     1%       /var/run
```

由以上输出可以看出，sda7 中共有 610800 个 i 节点，并且已经使用了 29%。

df 还可按照分区的类型显示文件系统。这个特性可以用于显示特定类型的分区使用情况。如下的命令显示了当前系统中所有 ext3 类型的分区的使用情况。为了对照，后面还输出了全部分区的相关信息，读者可以进行比较，深入理解 df 命令的-t 参数。

```
/dev/disk $ df -t ext3                  #显示类型为 ext3 的所有文件系统
文件系统      1K-块        已用          可用          已用%    挂载点
/dev/sda7    9614116     4249112     4876632     47%      /
/dev/sda6    124427      29665       88338       26%      /boot
/dev/sda9    61896904    41101684    17675804    70%      /home
/dev/disk $ df                          #不带参数,则显示普通文件系统的使用情况
文件系统      1K-块        已用          可用          已用%    挂载点
/dev/sda7    9614116     4249112     4876632     47% /
tmpfs        1031664     4           1031660     1%       /lib/init/rw
varrun       1031664     216         1031448     1%       /var/run
varlock      1031664     0           1031664     0%       /var/lock
udev         1031664     2796        1028868     1%       /dev
tmpfs        1031664     764         1030900     1%       /dev/shm
lrm          1031664     2000        1029664     1% /lib/modules/2.6.27-10
-generic/volatile
/dev/sda9    61896904    41101264    17676224    70%      /home
```

以上输出格式与前例的格式基本相同，这里就不再详细说明了。读者可自行分析其输出的意义。

9.3.5 使用 du 查看目录信息

如果要查看一个目录中所有文件占用的总空间，可以使用命令 du（disk usage）。该命令可以显示一个目录中每一个子目录所占用空间的大小，也可以显示每一个文件的大小等信息。该命令的使用格式是：

```
du [选项] [目录…]
```

可用选项如下所示。

- -s：只显示该目录中占用空间的总数。
- -a：递归显示各层目录中所有文件的大小。
- -b：以 B 为单位显示磁盘使用情况。

如果 du 命令不带任何参数使用，则显示当前目录及各子目录的空间使用情况，命令使用及输出如下所示。

```
du       #使用不带参数的 du 命令显示当前所有目录及子目录的空间使用情况
4    ./ZD1211LnxDrv_2_21_0_0/.tmp_versions
20   ./ZD1211LnxDrv_2_21_0_0/Winevl_iface
…
#第一列的数字是后面目录所占用的空间的块(每个块 1024B)大小
16   ./fcitx-svn/.svn/tmp
4    ./fcitx-svn/.svn/props
4    ./fcitx-svn/.svn/prop-base
4    ./fcitx-svn/.svn/text-base
40   ./fcitx-svn/.svn
./fcitx-svn
2595828                          #最后一行是当前目录所共占用的空间大小,以块为单位
```

以上输出说明./fcitx-svn 目录占用了 25958238 个块的空间。其中，/fcitx-svn 目录占用了 44 个块，./fcitx-svn/.svn/tmp 占用了 16 个块等。

在很多时候，显示每个目录占用的空间是没有什么意义的。因为在大多数情况下，我们只是想知道某目录及其所有子目录占用的所有空间总和，也就是说只要上个例子最后一行的内容即可。这时就可使用-s 参数显示总空间的占用情况，如下所示。

```
du -s /bin
/bin  #使用带有 s 参数的 du 命令，则只输出该目录共占用的磁盘空间。不列出明细表
```

由以上输出可知，/bin 目录所占用的空间大小是 6150 块。

9.4　hdparm 查看及设置磁盘信息

hdparm 在磁盘管理中也是一个极为重要的命令，其提供了查看及设置磁盘各种参数的功能。下面通过一些例子来说明 hdparm 常用的设置操作。

磁盘传输速度是磁盘的一个重要技术指标，其直接关系到磁盘的性能。hdparm 可以用来测试磁盘的传输速度。如下命令可检测 sda 磁盘的传输速度。

```
~/Documents $ sudo hdparm  -tT /dev/sda            #测试磁盘传输速度
/dev/sda:
 Timing cached reads:1662 MB in  2.00 seconds = 831.44 MB/sec
 Timing buffered disk reads:  104 MB in  3.06 seconds =  34.04 MB/sec
#如上输出说明磁盘 sda 的平均传输速度是 34.04MB/sec
```

磁盘的 32 位 I/O 支持也很重要。hdparm 使用参数 c 查询磁盘是否打开 32 位 I/O 支持。示例如下：

```
~/Documents $ sudo hdparm -c /dev/sda      #查询磁盘是否打开 32 位 I/O 支持。
/dev/sda:
 IO_support =  0 (default)
#如上输出说明磁盘并未打开 32 位支持
```

硬盘可以有几种能耗模式，如活动状态、休眠状态等。使用参数 C 可能查看其模式。示例如下：

```
~/Documents $ sudo hdparm -C /dev/sda              #检查磁盘的能耗模式
/dev/sda:
 drive state is:  active/idle
#如上输出说明磁盘处于活动状态
```

如果使用不带参数的 hdparm，则会输出许多有用的信息，包括是否打开 32 位 I/O 支持及硬盘的大小、柱面信息等。示例如下：

```
~/Documents $ sudo hdparm  /dev/sda
/dev/sda:
 IO_support =  0 (default)
 readonly=  0 (off)
 readahead  = 256 (on)
 geometry= 12161/255/63, sectors = 195371568, start = 0
#如上输出显示了磁盘的一些基本信息,如磁盘支持 32 位 I/O、未设置为只读模式等
```

使用 g 参数可以获得相关的物理信息，如柱面、磁头等信息。示例如下：

```
~/Documents $ sudo hdparm  -g /dev/sda
#显示磁盘物理位置(柱面,磁头,扇区),磁盘大小等信息。
/dev/sda:
 geometry= 12161/255/63, sectors = 195371568, start = 0
```

使用 i 参数得到的信息比较多，基本包括了所有磁盘相关的信息。示例如下：

```
~/Documents $ sudo hdparm  -i /dev/sda
#显示引导磁盘时所获得的较为详细的硬件信息
/dev/sda:
 Model=Hitachi HTS541210H9SA00  , FwRev=HPBOC20F, SerialNo=HP2D00BDG2EV5A
 Config={ HardSect NotMFM HdSw>15uSec Fixed DTR>10Mbs }
 RawCHS=16383/16/63, TrkSize=0, SectSize=0, ECCbytes=4
 BuffType=DualPortCache, BuffSize=7528kB, MaxMultSect=16, MultSect=?16?
 CurCHS=16383/16/63, CurSects=16514064, LBA=yes, LBAsects=195371568
 IORDY=on/off, tPIO={min:120,w/IORDY:120}, tDMA={min:120,rec:120}
 PIO modes:  pio0 pio1 pio2 pio3 pio4
 DMA modes:  mdma0 mdma1 mdma2
 UDMA modes: udma0 udma1 udma2 udma3 udma4 *udma5
 AdvancedPM=yes: mode=0x80 (128) WriteCache=enabled
 Drive conforms to: ATA/ATAPI-7 T13 1532D revision 1:  ATA/ATAPI-2,3,4,5,6,7
 * signifies the current active mode
```

```
#以上的输出内容最为详细,包括了磁盘信息的许多方面。如磁盘型号为 Hitachi HTS541210H9SA00、序列号为 HP2D00BDG2EV5A、缓冲区大小为 7528kB、开启了 LBA 支持等。一般用户不可能读懂所有输出信息,但只要记得该命令,在需要时使用,总可以找到想要寻找的磁盘的相关信息
```

hdparm 的参数很丰富，除了上面介绍的参数外，hdparm 还可使用其他参数，来获得关于磁盘的更为详尽的信息。这些参数如下所示。

- -d：设置磁盘的是否使用 DMA 模式。
- -m n：设置磁盘的多重扇区 I/O 值。可以设置为-m 16 等以提高磁盘性能（谨慎使用）。
- -r：获得或设置磁盘的只读标志，如果设置为只读，则无法向磁盘中写入数据。
- -W：打开/关闭磁盘的写缓存。
- -y：使磁盘进入待机模式。
- -Y：使磁盘进入睡眠模式。

9.5　磁盘内容复制

Linux 中有一个强大的磁盘内容读写软件，就是 dd。dd 可以完成磁盘内容的备份及恢复，而且还可以从内存、光盘等介质中读取数据；在复制数据的过程中，还可对数据进行相应的转换。dd 的主要选项如下所示。

- if=file：输入文件名，也就是 dd 要操作的源文件名。文件名可以是磁盘、光盘等介质，也可以是一个普通文件。
- of=file：输出文件名。指定 dd 命令将源文件输出到什么位置。
- ibs=n：一次读取 n 个字节。相当设置读取块的大小。
- obs=n：一次写入 n 个字节。设置写入块的大小。
- bs=n：读写块的大小。相当于同时使用了 ibs 与 nbs。
- cbs=n：转换缓冲区的大小。
- skip=n：从输入文件开头跳越 n 个块后再开始操作。
- seek=n：从输出文件跳越 n 个块后再进行操作。
- count=n：仅复制 n 个块。
- conv=参数：参数可取 ascii、ebcdic、ibm、lcase（转换为小写字母）、ucase（转换为大写字母）、swab 交换输入的每对字节。

仅仅看看参数，也是很难理解 dd 的行为特征的。所以，下面提供了比较丰富的示例，从备份磁盘数据到制作光盘 ISO 等，相信理解了示例文件，就会更为深刻地理解前述参数的意义。

- 将 EBCDIC 格式的文件（ebc_1.file）转换为 ASCII 格式文件（asc_1.file），代码如下：

```
dd if= ebc_1.file of = asc_1.file conv=ascii
```

- 将 sda1 分区备份到 sdb5，代码如下：

```
dd if=/dev/sda1 of=/dev/sdb5
```

- 将 sda1 分区备份到 sdb5/disk1.file 文件，代码如下：

```
dd if=/dev/sda1 of=/dev/sdb5/disk1.file
```

- 将 sda1 分区压缩并备份到 sdb5/disk1_image.gz 文件，代码如下：

```
dd if=/dev/sda1 | gzip >/dev/sdb5/disk1_image.gz
```

- 恢复 dd 所备份的文件，代码如下：

```
dd if=/dev/sdb5/disk1.file of=/dev/sda1
gzip -dc /dev/sdb5/disk1_image.gz | dd of=/dev/sda1
```

- 备份磁盘分区表 MBR，代码如下：

```
dd if=/dev/hda of=/dev/sdb5/disk1.mbr count=1 bs=512
```

- 恢复磁盘分区表 MBR，代码如下：

```
dd if=/dev/sdb5/disk1.mbr of=/dev/hda
```

- 复制内存资料到磁盘，代码如下：

```
dd if=/dev/mem of=/dev/sdb5/mem.file bs=1024
```

- 将光盘文件制作为 iso 镜像，代码如下：

```
dd if=/dev/cdrom of=/sdb5/Ubuntu.iso
```

- 用随机数据填充 sdb5，以彻底销毁磁盘数据，代码如下：

```
dd if=/dev/random of=/dev/sdb5
```

- 修复磁盘，代码如下：

```
dd if=/dev/sda of=/dev/sda
```

9.6 小结

本章介绍了 Linux 磁盘管理的相关知识，如使用 Live CD 中的 Gparted 进行磁盘管理（如分区、格式化等），还介绍了磁盘管理的命令行程序，以及查看磁盘使用情况的几个实用程序。本章的重点内容是“用 Live CD 进行磁盘管理”，该节主要介绍如何在裸机上使用 Live CD 进行磁盘分区、格式化的方法，这些功能在很多情况下是很实用的。

第 10 章　进 程 管 理

进程就是计算机中运行的程序。管理进程的工作主要包括查看进程占用的资源、启动及杀掉进程，以及调整进程的运行状态、自动运行进程等。进程管理在计算机管理中有着极其重要的位置，掌握了系统中运行的进程以及进程所占用的系统资源后，才能有效地调整进程，以使系统稳定地运行。

10.1　进程知识简介

要有效地管理进程，首先应该对进程有个概念上的认识。本节简单介绍了进程相关的知识及守护进程、僵尸进程的概念。

10.1.1　进程的基本概念

进程就是运行在 Linux 中的程序，用户所运行的每个程序均称为一个进程。进程名就是程序的名字。进程在 Linux 操作系统中是内核管理的基本单位。

进程运行后，会占用一定的系统资源，如 CPU 的运算资源、系统内存、系统寄存器等。从系统管理的角度来看，进程结构可以分成三部分：用户级数据、寄存器数据、系统级数据。用户级数据是进程的主要部分，包括程序数据、用户堆栈等；寄存器数据区包括了指令寄存器及其他程序运行必需的寄存器信息；系统级数据区是指系统管理所需的数据，如进程页表、task-stuct 结构等信息。

Linux 在管理进程时，主要是依赖 task_struct 数据结构，该结构描述了当前系统中运行的所有进程的状态及相关资源状况。在 Linux 内核中，有一个 task_struct 数组，系统中运行的每一个进程均对应一个 task_struct 结构。管理进程主要就是对 task_struct 进行操作。



在 Ubuntu8.04 版本中，对进程而言极其重要的数据结构 task_stuct 的定义在源代码“/usr/src/linux-headers-2.6.24-16-generic/include/linux/sched.h”的 920 行左右处。其结构较为复杂，包含了很多方面的内容，在源代码中有 200 多行。有兴趣的读者可以自行查看。

Linux 中，进程不能被从零开始创建，所有的进程均是使用 fork 函数从 init 进程复制而得来的。可以使用 pstree 来验证一下。在笔者的系统中，使用 pstree 得到的部分输出如下所示：

```
init─┬─NetworkManager
     ├─dhcdbd
     ├─dhclient3
     ├─gconfd-2
     ├─gdm─────gdm─┬─Xorg
     │   └─fvwm2─┬─2*[FvwmButtons]
```

```
|  ├─FvwmIconMan
|  ├─FvwmPager
|  ├─FvwmScript
|  ├─emacs─────bash─────pstree
```

可以看到，所有的进程均是 init 的子进程。在 init 的子进程 gdm 中，又派生出了子进程。而这些结构，均记录在 task_struct 数据结构中。

10.1.2 守护进程

守护进程是仅运行在后台，不与用户执行交互操作的一些进程，一般以字母 d 作为名称的结尾。如 klogd（核心日志守护进程）、atd（自动调度守护进程）等。

大多数用户调用的进程如各种 Shell 进程，其完成的工作依赖于与用户的交互：用户输入内容，其执行，然后输出相应的内容反馈给使用者。但有些服务是不需要与用户进行交互的，如系统日志的记录，由于日志产生的速度极快而且数量极多，如果与用户交互等待用户的命令再记录日志，很明显是来不及的。这时，就需要使用守护进程，在相应的事件发生后，自动将事件的相关内容记录在日志中，以供用户查看。还有 HTTP 服务器、邮件系统等都是这样的原理，所以将诸如此类的进程设计为守护进程，自动在后台完成相应任务。

当然，一般情况下守护进程不与用户交互，并不代表其永远独立运行，无法对其进行控制。Linux 系统提供了一种称为信号的操作，可以与其进行通信。

10.1.3 僵尸进程

父进程退出后，子进程由于某种原因却没有被父进程正常回收，这时失去了父进程的子进程就称为僵尸进程（zombie）。

进程在调用 exit 函数退出时，其实并没有真正从系统 task_struct 数组中被销毁，而只是将普通进程变成一个僵尸进程。这时，进程释放了其所占用的系统资源，如其执行代码、内存空间等，只保留了进程的退出状态供收集之用。如果该进程的父进程没有完成子进程的清除工作，则该子进程会被提交给 init 进程统一管理。但如果父进程陷入一个循环之中，即没有完成子进程的清除工作，又不能及时将子进程交给 init 来管理，那么子进程就会一直陷入僵尸状态。

僵尸进程的处理方法是杀掉父进程。出现僵尸进程后，只要杀掉该进程的父进程，就可以将该进程提交给 init 进程，init 进程会完成清理僵尸进程的工作。具体做法只需三个步骤：

（1）用 ps -le 等命令发现僵尸进程（带有标记 Z 的进程就是僵尸进程）；

（2）用 pstree 等命令来查看进程树，找到僵尸进程的父进程；

（3）用 kill pidnum（僵尸进程的父进程的 PID 号）或 kill -9 pidnum。

10.2 进程监视

要对进程进行管理，首先应该知道系统中运行进程的基本情况，如进程的名称列表、资源占用等情况。Linux 提供了丰富的用于监视进程性能的工具，有静态显示进程列表的 ps、memstat 等命令，也有动态显示进程占用资源的 top 等命令。本节就将介绍这些工具。

10.2.1　进程显示命令 ps

进程显示是进程管理中最为基本的一个环节。其功能就是显示系统中运行进程的详细情况，为了解、分析系统情况提供依据。Linux 系统中，最常用的显示进程的命令就是 ps。

ps 相对来说是一个比较复杂的命令，使用中有许多需要注意的地方。同时其参数的使用也是一个复合的整体，所以这里就不一一列举参数的意义，而是用一些实例说明这些参数复合后其输出的内容及其功能。

1．显示当前系统中所有的进程

显示当前系统的所有进程是比较常用的功能，ps 可以使用各种参数来控制显示进程的不同信息。具体的可使用参数及其部分输出如下所示，其输出的含义可通过查看本节第 8 条来理解，这里就不再进行详细解释了。

```
westnorth@18:58 ~$ ps -e |head -n 2
#输出当前系统所有进程，用 head 命令控制只输出两个记录
  PID TTY TIME CMD
 1 ?  00:00:01 init
westnorth@18:59 ~$ ps -ef |head -n 2
#添加 f 参数的，可以显示进程的更为详细的信息，这些显示项的意义，将在后文予以解释
UID  PID  PPID  C STIME TTY TIME CMD
root1  0  0 11:22 ?  00:00:01 /sbin/init
westnorth@18:59 ~$ ps -eF |head -n 2
#如果使用 F 参数，则显示的信息就包含更多的内容
UID  PID  PPID  C SZRSS PSR STIME TTY TIME CMD
root1  0  0711  16881 11:22 ?  00:00:01 /sbin/init
westnorth@18:59 ~$ ps -ely|head -n 2
#使用 ly 参数，显示了一些状态信息
SUIDPID  PPID  C PRI  NIRSS SZ WCHAN  TTY TIME CMD
S  0  1  0  0  800  1688711 -?  00:00:01 init
```

2．输出进程树

为了查看进程的继承关系，可能需要输出进程树。ps 提供了输出进程树的参数，可以这样使用 ps。

```
westnorth@19:03 ~$ ps axjf
#使用这几个参数，就会看到进程以树的形式被显示
 PPIDPID  PGIDSID TTYTPGID STATUIDTIME COMMAND
...
    1 17243  5635  5635 ?          -1 S    1000   0:09 mlterm
17243 17244 17244 17244 pts/1    13047 Rs    1000   0:00  \_ bash
17244 13047 13047 17244 pts/1    13047 R+    1000   0:00      \_ ps axjf
17244 13048 13047 17244 pts/1    13047 S+    1000   0:00      \_ less
    1 22687  5635  5635 ?          -1 Sl    1000   0:24 stardic
```

由输出可以看到 mlterm 与 stardict 是同级别的进程，而 bash 是从 mlterm 继承而来的，同时 bash 又派生出了 ps axjf 与 less 进程。

3．获得线程信息

为了查看更为详细的资源占用以及其他信息，可以通过查看线程来实现。具体的实现方法如下：

```
westnorth@19:09 ～$ ps axms
  UIDPID PENDING BLOCKED IGNORED  CAUGHT STAT TTY  TIME COMMAND
 0  1 0000000000000000 - - - - ? 0:01 /sbin/init
 0  - 0000000000000000 0000000000000000 0000000000001000 00000000280b6423
Ss- 0:01 -
 0  2 0000000000000000 - - - - ? 0:00 [kthreadd]
0  - 0000000000000000 0000000000000000 ffffffffffffffff 0000000000000000
S<- 0:00 -
```

输出显示的内容涉及系统底层信息，将其列出是为了展示 ps 命令的功能，普通用户不必理解其具体意义，这里就不再介绍了。

4．获得进程状态信息

接触过《操作系统》的读者可能知道，Linux 系统的进程有十几种不同的状态。而 ps 命令可以将进程的不同状态显示出来。具体参数如下：

```
westnorth@19:13 ～$ ps axZ | head -n 5
LABEL  PID  TTY STAT    TIME    COMMAND
-       1   ?   Ss      0:01    /sbin/init
-       2   ?   S<      0:00    [kthreadd]
-       3   ?   S<      0:00    [migration/0]
-       4   ?   S<      0:00    [ksoftirqd/0]
```

显示结果说明这些进程均处于睡眠状态，并且除了 init 进程外，其余进程均处于高优先级状态。

5．查看特定用户的进程信息

查看用户名为 westnorth 的用户的详细信息可以使用如下命令，其输出如下：

```
westnorth@19:21 ～$ ps -l -U westnorth|head -n 6
F SUIDPID  PPID  C PRI  NI ADDR SZ WCHAN  TTY TIME CMD
4 S  1000  5630  5388  0  800 -  5664 -?  00:00:09 fvwm2
1 S  1000  5702  1  0  800 -  4313 -?  00:00:30 scim-launcher
1 S  1000  5719  1  0  800 -  1037 -?  00:00:00 scim-helper-man
1 S  1000  5720  1  0  800 -  8500 -?  00:00:48 scim-panel-gtk
1 S  1000  5722  1  0  800 -  1703 -?  00:00:23 scim-launcher
```

输出显示了属于用户 westnorth 的进程有 fvwm2、scim-launcher 等。

6．查看特定名称的进程

也可以将 ps 命令当成一个查询器，用于查询特定名称的进程的 PID 号。使用方式如下：

```
westnorth@19:44 ～$ ps -C kthreadd -o pid=
2
```

输出显示 kthreadd 的进程 ID 号是 2。

7．查看特定 PID 号的进程

查看 PID 号为 7855 是哪一个进程可用如下参数：

```
westnorth@19:47 ~$ ps -p 7855 -o comm=
mlnet
```

输出显示 PID 号为 7855 的进程是 mlnet。

8．部分输出参数的意义

可以参照上面的输出理解。

- C：CPU 占用指数，指数越大，则说明占用 CPU 越多。
- USER：进程属主用户名。
- UID：用户 ID 值。
- PID：进程 PID 号。
- PPID：父进程的 PID 号。
- %CPU：进程 CPU 使用率。
- %MEM：进程内存占用率。
- VSZ：进程占用的虚拟内存大小。因为系统中还有一些共享库，VSZ 就是进程使用的内存加上该进程使用的共享库占用的空间总和。
- RSS：进程使用的物理内存大小。
- TTY：终端号码。
- S：与 STAT 相同。
- STAT：行程的状态，状态又有如下选项。
 D：不可中断。
 R：运行状态。
 S：睡眠状态。
 T：暂停状态。
 Z：僵尸进程。
 W：内存不足。
 <：优先级别高。
 N：优先级别低。
- START：进程开始运行的时间，用这个可以判别进程运行的时间。特别地，可以查看 init 的开始时间来查看系统运行时间。
- STIME：进程开始运行的时间。
- SZ：进程的内存占用数量，以页为单位。如果该值一直不断增长，则可判断进程发生了内存泄露。
- TIME：进程被 CPU 执行的时间。
- PRI：进程的优先级，数值越小，优先级越高。
- NI：进程的 nice 值，nice 值是表示系统优先级的静态系统变量。
- WCHAN：进程在睡眠前发出的信号值，用以表示唤醒该进程所需的操作。
- COMMAND：进程实际执行的程序名。

9．其他参数

- -A：显示所有进程。
- -w：显示宽格式进程信息，也就是每行可以显示超过 80 个字符宽度的内容。

10.2.2 memstat 显示共享库的内存占用

memsat 命令的功能是显示共享库所占用的内存，这些内存只是被声明使用，其实不一定被实际使用。所以该命令显示的内存使用量会很庞大。其使用很简单，直接在终端中输入 memstat 就可以，如果加上-w 参数，memstat 就不会把每行显示的内容截断到 80 列（将内容截断是默认行为）。

```
westnorth@20:41 ~$ memstat -w
2792k: PID  5630 (/home/westnorth/.fonts/ZhunYuan.ttf)
  10804k: PID  5702 (/usr/lib/locale/zh_SG.utf8/LC_CTYPE)
 292k: PID  5719 (/lib/tls/i686/cmov/libpthread-2.7.so)
  10472k: PID  5720 (/SYSV00000000)
 428k: PID  5722 (/usr/lib/libXdmcp.so.6.0.0)
…
52k: /home/westnorth/.cache/stardict/langdao-ce-gb.idx.oft 9752
4k: /home/westnorth/.cache/stardict/hanzim.idx.oft 9752
  12k: /home/westnorth/.cache/stardict/jmdict-en-ja.idx.oft 9752
8k: /home/westnorth/.cache/stardict/dictd_www.dict.org_gazetteer.idx.oft 9752
--------
1880412k
```

输出的内容中，第一列显示的 2792k 之类的数字是该共享库的内存占用。显示分为两部分，前半部分在进程的内存占用后紧跟着就是使用该共享库的进程 PID 号，最后才是共享库的完整路径。而后半部分，共享库的内存占用后，先是为内存占用，随后是共享库的完整路径，最后才是使用这个共享库进程的 PID 号。查看输出，可以发现，有许多进程使用同一个共享库，也许多共享库被同一个进程使用。

可以给共享库占用的内存数量排序，以查看哪一个共享库占用内存最多。方法很简单，只要将 memstat 的输出重定向到 sort 命令中即可，如下所示。如果不明白命令的意思，请见第 6 章，查看 6.3.2 节 sort 命令相关内容。

```
memstat -w |sort -t 'k' -rn
```

10.2.3 pstree 以树型列表方式显示进程

在介绍 ps 的参数时，提到可以使用 axjf 参数来显示进程树。但实际使用就可以知道，该参数显示的进程树是相当粗糙的。所以，Linux 提供了一条专门用于显示进程树的命令 pstree。其使用很简单，直接在终端中输入 pstree 就可以了，其输出如下：

```
westnorth@21:26 ~$ pstree
init─┬─NetworkManager
  ├─gdm─────gdm─┬─Xorg
  │   └─fvwm2─┬─2*[FvwmButtons]
  │  ├─FvwmIconMan
  │  ├─emacs─────bash─────pstree
  │  ├─FireFox─────6*[{FireFox}]
  │  ├─mlterm─┬─bash─┬─WINWORD.EXE─┬─rpcss─────3*[{rpcss}]
  │  │   │ │  └─7*[{WINWORD.EXE}]
```

```
    |  |  |  └─mlnet─────2*[{mlnet}]
    |  |  ├─bash
    |  |  └─bash─────man─────pager
    |  ├─seahorse-agent
    |  ├─ssh-agent
    |  └─stardict
```

由输出可以看到，init 是根进程，派生了 gdm，而 gdm 又派生了 mlterm，mlterm 又派生了 bash，bash 派生了 man 等。其他进程之间的关系可以自行分析。

pstree 还可以使用参数来控制其输出。下面是两个简单的例子。

```
pstree westnorth     #显示用户 westnorth 的进程树
pstree 10223         #显示以 PID 号 10223 为根的进程树
```

还有一些特殊的选项，如下所示：

```
pstree -p            #显示进程的 PID 号
pstree -u            #显示用户 ID
pstree -n            #以父进程的 PID 号进行排序输出
```

10.2.4　进程筛选命令 pgrep

pgrep 可以对进程按照各种条件进行筛选。相对来说，其输出的内容不很丰富，默认只输出进程号。但其作用不在于提供丰富的进程信息，而是对进程进行筛选，从中选择最符合条件的而已。看下面的使用实例。

```
#查找系统中属于用户 westnorth 的进程,同时该进程的名称是以 mlter 开头
westnorth@23:19 ～$ pgrep -f 'mlte*'  -u westnorth -l
#得到的输出是进程 PID 与进程名
10722 mlterm
```

其命令格式如下：

```
pgrep [参数] -u 用户名
```

这个命令在执行时，必须提供用户名，才能显示相关的进程信息。其可用参数列表如下所示。

- -c：只统计进程的总数而不输出进程的列表。
- -d：使用新的定界符来取代默认的回车符。
- -f：匹配包含命令行在内的内容，默认只匹配进程名。
- -g：只匹配特定的组中的进程。
- -l：列出进程名称，默认情况下只列出进程号。
- -n：只列出所有匹配项中最新的一项，也就是启动时间最晚的项。
- -o：只列出最旧的一项，也就是运行时间最长的一项。
- -p：只列出父进程被列出的项。
- -u：只列出用户名被列出的项。
- -v：列出不匹配的项。
- -x：只列出与输入项完全匹配的进程。

10.2.5 动态监视进程命令 top

10.23 节和 10.2.4 节介绍的命令均是静态显示进程的命令，由于系统中的进程所占用的系统资源等均是动态变化的，要更准确地查看系统中进程的动态运行情况，就应该使用动态进程监视程序。动态监视进程的程序有 top 与构建在 top 基础上的 htop。

top 是一款强大的动态显示进程的资源占用的软件。其功能强大，除了可以动态显示进程的资源占用外，还显示了进程的状态统计信息、内存占用总量等信息，其功能界面如图 10.1 所示。

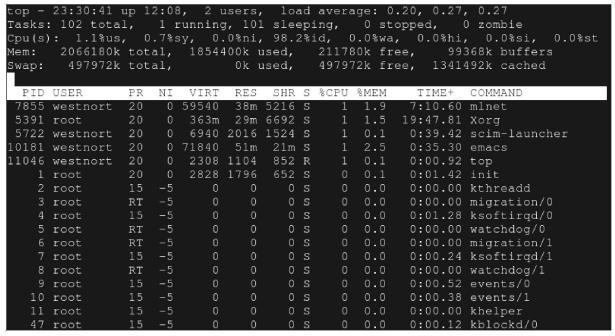


图 10.1　top 运行界面图

top 命令最简单的用法就是直接在终端中输入 top 指令即可，其运行界面如图 10.1 所示。top 是一个以 CPU 占用率排序的动态进程显示程序。其显示的内容很多，默认每 3 秒钟更新一次。显示的各项意义如下所示。

- up：依次显示系统已经运行的时间（本例是 12：08，意思是已经运行了 12 小时 8 分钟）、登录到本系统的用户数（本例为 2）、平均负载情况，依次是最近 1 秒（本例显示为 0.20，系统负载为 0.20%）、5 秒（本例显示为 0.27）、15 秒（本例显示为 0.27）的负载值。
- Tasks：系统进程统计表，依次是进程总数（102，意思是共有 102 个进程正在运行，下同）、正在运行的进程（1）、休眠的进程（101）、停止的进程（0）僵尸进程（0）。
- CPU：显示各种模式中 CPU 的占用率，用户模式（us），系统模式（sy），优先级进程（ni）等。
- Mem：内存使用量。依次是总内存（total），已用内存（used），空闲内存（free），缓存内存（buffer）。
- Swap：交换分区使用量，交换分区相当于 Windows 系统中的虚拟内存。交换分区显示内容与内存基本一致。
- PID：进程的 PID 号。
- USER：进程属主名。
- PR：进程优先级。
- NI：进程静态优先级值。
- VIRT、RES、SHR：内存使用情况。
- S；进程状态。R 为运行，S 为休眠。
- %CPU：进程最近所占用 CPU 时间的比例。
- %MEM：进程占用的物理内存占总内存的比例。

- TIME：进程启动以来所占用的总 CPU 时间。
- COMMAND：进程的实际程序名称。

top 命令在运行中，用户还可与命令进行简单的交互操作，以控制其显示。这些交互操作的命令一般是一个字母，意思是按一个字母就可进行相应的操作。这些操作及其作用如下所示。

- space：空格，立即刷新。
- h 或?：显示帮助信息，包含这里介绍的所有命令。
- B：是否将某些内容以粗体显示。
- k：终止进程。输入 k 后，命令会出现一个提示输入文本的区域，在其中输入要终止进程的 PID 号即可。
- i：显示 / 不显示休眠和僵尸进程。开关式命令，按一次显示，再按一次则不显示休眠与僵尸进程。
- q：退出。
- r：重置一个进程的优先级。系统会提示先输入进程的 PID 号，然后再输入优先级。
- s：重置刷新时间间隔，默认是 3 秒。直接在系统提示输入时间间隔处输入新的时间间隔数即可。
- f 或 F：更改显示项，可以自行设置让 top 命令显示那些项目。
- o 或 O：更改显示项目的顺序。
- l：是否显示 uptime 内容。
- H：是否显示线程。
- m：是否显示内存信息。
- t ：是否显示进程和 CPU 状态信息。
- c：是否显示命令名称和完整命令行。
- M：根据进程的内存占用大小进行排序。
- n：显示进程数，0 为显示所有进程。
- R：反序显示。
- P：根据 CPU 使用率进行排序。
- T：根据进程使用 CPU 的时间进行排序。
- W：将当前设置写入配置文件～/.toprc，这样下次启动时就会按照此设置显示。
- x：高亮显示排序的列。
- y：高亮显示当前进程。
- Z：更改色彩显示模式。

top 命令在启动时也可以指定某些参数来定制其行为模式，大多数参数均可在 top 命令的交互模式中设置，就不再介绍，只介绍两个不能在交互模式中使用的参数。

- n：指定刷新的次数，到次数限制后，top 会自动退出。
- b：批处理模式，与 n 参数一起使用，可以用来将 top 的显示内容写到一个文件中。

非交互式使用 top 的示例，其功能是更新两次系统进程信息，然后将这些内容输出到文件 session.log 中，注意 session.log 必须已经存在，示例如下：

```
top -n 2 -b < session.log
#查看该文件，其部分内容输出如下：
```

```
top - 10:34:49 up  2:42,  1 user,  load average: 0.17, 0.30, 0.33
Tasks: 100 total,1 running,  99 sleeping,0 stopped,0 zombie
Cpu ㈤ :  4.8%us,  2.0%sy,  0.0%ni,92.3%id,  0.8%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:2066180k total,992920k used,  1073260k free,  75944k buffers
Swap:497972k total,  0k used,497972k free,537676k cached
  PID USERPR  NI  VIRT  RES  SHR S %CPU %MEM TIME+  COMMAND
 6264 westnort  200  271m 113m  26m S 2  5.63:35.98 FireFox
 1 root200  2844 1688  544 S 0  0.10:01.40 init
 2 root15  -5  0 0 0 S 0  0.00:00.00 kthreadd
 3 rootRT  -5  0 0 0 S 0  0.00:00.00 migration/0
 4 root15  -5  0 0 0 S 0  0.00:00.14 ksoftirqd/0
…
```

10.3 终止进程

如果一个进程的运行导致了非预期的结果，如程序的停止响应；或者程序的使命完成，已经没有必要存在系统中，在这些情况下，就需要将进程结束。结束进程，一般使用的命令是 kill。在介绍 kill 命令前，先应该了解一下 Linux 系统中关于信号的概念。

10.3.1 信号的基本概念

信号是打断进程运行的“中断”。系统中有许多中断，如断电导致的硬中断，会强制系统中所有硬件停止运行；也有软件发出的“除 0”中断，如果软件在计算中发现除数是 0，也会发出一个中断信号，终止程序的运行。信号也是中断的一种，其作用是将某些特殊的控制命令或消息发送给进程。在 Linux 系统中可用 kill -l 查看信号列表。下面将一些常见信号及其意义列表如下，括号内的数字为其信号值。

- SIGHUP（1）：挂断。如果进程是由一个终端启动，则终端退出，则进程会收到挂断信号。
- SIGINT（2）：中断。用户按下“Ctrl+c”键产生的信号。
- SIGQUIT（3）：退出。用户按下“Ctrl+\”键产生的信号。
- SIGILL（4）：非法硬件指令。
- SIGTRAP（5）：陷阱信号，可用于程序调试。
- SIGFPE（8）：被 0 除或发生浮点数异常时的信号。
- SIGKILL（9）：强制终止信号，进程接到该信号后，会无条件终止运行。
- SIGUSR1（10）：用户自定义信号。
- SIGSEGV（11）：段违例，如果用户进程访问了系统核心内存，会引发该信号。
- SIGPIPE（13）：进行管道操作的两个进程，如果负责接收的进程关闭了管道，负责发送的进程会收到此信号。
- SIGALRM（14）：定时器时间到后发出的信号。
- SIGTERM（15）：软件终止信号，是 kill 命令默认发送的信号。
- SIGTSTP（20）：挂起操作。

10.3.2　kill 命令

kill 命令的作用是向进程发送特定的信号，以完成不同的功能，如终止进程、重启进程等。其最简单的使用方法如下：

```
#终止 PID 为 10232 的进程。没有显式指定信号量，系统默认发送信号量为 15（TERM）的信号
#下列所有的命令功能相同
kill 10232
kill -15 10232
kill -s 15 10232
kill -s 15 -p 10232
kill -SIGTERM 10232          #最后两个命令发送的信号量并不是 15，而是 9
kill -  TERM 10232           #但最终导致的结果是相同的，均为终止进程
```

由上例可以看到，指定发送到进程的信号量有许多等效的方式，-15、-s 15、-SIGTERM、- TERM。一般常用的信号量有 HUP（1）、KILL（9）、TERM（15），其意义可查前面列出的表得到。当然，kill 命令发送的信号远不止这些，基本上前面列出的信号量均可发送给进程，大多情况下导致进程退出。下面是一些常用信号的意义。

- HUP：可以使用-1、-HUP、-SIGHUP 三种格式发送给进程。意思是让进程重新启动。如 kill -HUP 10232 就可以让 PID 为 10232 的进程重新启动。这在修改了其配置文件时比较有用。
- KILL：这是系统默认发送的信号量。同样可用-15、-KILL、-SIGKILL 等方式发送给进程。作用是发送中止信号给进程，这是非强制的方式，进程有充足的时间用来释放所占用的系统资源。是比较安全的退出方式。
- TERM：同样可以使用-9、-TERM、-SIGTERM 三种格式发送给进程。这是强制中止的信号，无论进程正在做什么，接到此信号后必须强制退出，是无法屏蔽的信号。但这种方式终止进程会造成资源回收不到位以及其他问题，所以不是极其需要，一般不使用该信号终止进程。

前面介绍了 kill 的一些使用方法。但对于如何中止特定的进程，可能有的读者还是心存疑惑。现在介绍一下 kill 的使用技巧，如中止特定的进程及重启进程的方法。

- 使用前先用 ps aux 等命令列出所有进程，从中找到要终止进程的 PID 号。然后再使用下列命令执行相应的操作。
- 终止特定的进程：kill 2390 3020 12902。意思是终止 PID 号为 2390、3020、12902 的进程。
- 强行终止特定的进程：kill -9 2390 2891。意思是终止 PID 号为 2390 与 2891 的进程。注意该过程是强制性的，这两个进程在接到信号后，会马上终止。
- 重启特定的进程：kill -1 2390 12903。发送信号 1 给进程，意思是让进程重新启动。但要注意的是，这种信号一般是发给系统服务进程的，普通进程如 vim 等收到此信号只会退出。

10.3.3　按进程名终止进程 pkill

使用 kill 命令有点不方便之处，也就是 kill 只能终止指定进程号的进程，而普通用户在使用计算机时，一般只会记得进程名，如 vim、FireFox 之类。这样在使用 kill 时，必须先使用 ps 命令先列出进程的 PID，比较麻烦。为了方便这些只记得进程名称的用户，Linux 提供了一个

命令 pkill，可以直接使用进程名称终止进程。如可以这样使用：“pkill -15 vim”，就会终止名为 vim 的进程。pkill 的功能还远不止于此，其提供了强大的对于进程的筛选功能。完整的 pkill 命令格式如下：

```
pkill [参数] [模式]
```

pkill 的参数如下所示。

- -f：匹配包含命令行在内的内容，默认只匹配进程名。
- -g：只匹配特定的组中的进程。
- -n：只列出所有匹配项中最新的一项，也就是启动时间最晚的项。
- -o：只列出最旧的一项，也就是运行时间最长的一项。
- -u：只列出指定用户名的进程。
- -v：列出不匹配的项。
- -x：只列出与输入项完全匹配的进程。

下面是一些使用的实例。

```
pkill vim                    #终止名为 viim 的进程
pkill -HUP syslogd           #向进程发送 HUP 信号，意思是要该进程重启。这个选项
                             #一般用于守护进程，或一些系统进程，普通过程如 vim 等
                             #无法使用该重启指令
pkill win*                   #终止所有以 win 开头的进程
```

10.4 进程高级管理

仅仅知道如何列出进程状态和终止进程，对于系统管理来说是远远不够的。进程的管理，还至少应该包括将某些进程置于后台运行、调整进程的优先级以及使用进程的自动调度功能等。本节就将介绍这方面的内容，阅读完本节后，用户应该初步具备进程管理方面的高级技巧。

10.4.1 进程的后台执行

在一个终端中输入命令，运行程序，这是大多数用户均能熟练进行的操作。但是用户有时会有特殊的需求，如要在终端中打开 Emacs，接着还要打开 GIMP 等其他程序。如果像平常一样操作，在终端输入 Emacs 后终端就被 Emacs 独占，要运行 GIMP，就不得不再开一个终端。这样做，不但浪费系统资源，而且造成了管理上的麻烦。

想象一下，要运行五六个程序就需要五六个终端，在任务切换时就会麻烦很多。还有一种情况，如要运行某些复杂的计算任务，而这些计算工作是无须用户输入任何信息，但需要耗费很长时间。如果只是简单地在终端中运行该进程，则在漫长的计算中，无法在终端中做任何其他工作，这对于实际的工作是不可想象的。

在这些时候，就可以使用一种称为后台运行的技术，让多个进程同时运行于一个终端中。如果预料到进程的执行时间过长或要在同一个终端中执行另一个进程，则可以将该进程置于后台运行。这样就可以在后台运行比较耗费时间的进程，然后在终端中可以继续执行其他指令。将进程置于后台运行，有几种方法，现分别予以介绍。

1．进程后台运行符&

最简单的让进程后台运行的方法就是在终端中输入程序时，在最后加上&符号，就可以让该进程在后台运行。可以做如下的体验。

（1）在终端中输入任意程序名，如 gedit，可以观察到文本编辑器被运行。

（2）现在终端中显示的内容是文本编辑器运行时的状态信息。此时，终端被 gedit 独占，用户无法再输入其他命令。

（3）退出 gedit。

（4）在终端中输入 gedit&（注意其后的&符）。同样，文本编辑器开始运行。

（5）此时，再观察终端，终端中只输出了 gedit 的进程号，然后就回到了命令模式，用户可以输入命令再运行其他程序。

（6）可以在终端中输入 gvim，打开 vim-gtk 程序进行其他操作。

2．nohup 命令

使用“&”虽然简单易行，但却有个比较致命的缺陷，就是这样运行的进程的父进程仍然是当前的 Shell。如果 Shell 退出，则使用&运行的所有进程均会被强制退出，这有时会导致很严重的后果。为了解决这个问题，我们先来回顾一下有关信号的内容。

Shell 在退出时，会发送 HUP 信号给其所有子进程，而接到 HUP 信号的进程会中止运行。所以，只要忽略 HUP 信号，就可以使得进程在 Shell 退出时仍然保持运行。这就是 nohup 的工作原理。

nohup 命令的使用也很简单，直接将其加在要运行的程序之前即可。下面是一个用法示例：

（1）在终端中使用命令 gedit 打开文本编辑器；

（2）关闭终端，可以看到 gedit 随着 Shell 的关闭而被中止；

（3）在终端中使用命令 nohup gedit 打开文本编辑器；

（4）关闭终端，可以看到 gedit 并没有随着 Shell 的关闭而被中止。

但是 nohup 并不是一个让进程后台运行的命令，其只是让该进程忽略来自 Shell 的终止信号。使用 nohup gedit 后终端显示的仍然只是 gedit 的相关信息，并没有出现可以输入命令的提示符。

为了修正此问题，可以再使用一下&符号。像这样使用 nohup gedit &，就可以既让 gedit 在后台运行，又不让其随着 Shell 的退出而关闭。

nohup 后面跟的命令的标准输入被重定向到/dev/null，标准输出与错误输出被重定向到当前目录的 nohup.out 文件。关于标准输出、标准输入的概念，请参见本书 6.2.6 节的相关内容。如果当前目录没有写权限，就将标准输出重定向到用户的主目录。如果在用户的主目录也无法写入，nohup 程序就会终止运行。

3．setsid 命令

为了使得进程不被 Shell 中的 HUP 信号中断，nohup 提供的方法是忽略 HUP 信号，以使进程保持运行。还有一种方法也可以达到类似的目的，就是使用 setsid 命令。

HUP 信号是父进程发送给所有的子进程的信号，子进程接收到该信号后就会退出。所以，只要将某进程的父进程更改为系统的根进程，那么来自 Shell 的 HUP 信号就根本不会发送到该进程，这样也可以达到使进程保持运行的目的。这就是 setsid 的工作原理。setsid 命令的使用也很简单，直接加在命令之前即可。

```
setsid gedit
```

4．disown 命令

前面介绍的命令均是在程序运行时加上一定的参数，就可使程序在后台运行。但如果在程序已经运行的情况下，要使其在后台运行，就要使用另一个命令 disown 了，其使用方式一般有如下三种。

- disown -h name：使名称为 ame 的进程忽略 HUP 信号。
- disown -ah ：使所有的进程都忽略 HUP 信号。
- disown -rh：使正在运行的进程忽略 HUP 信号。

5．Ctrl+Z 挂起命令

还有一种比较简单的将进程放入后台运行的方法就是使用 Ctrl+Z 挂起命令。如果在程序正在执行的过程中，又需要执行其他命令，就可以使用 Ctrl+Z 将当前进程挂起，然后在终端中执行其他命令。要将挂起的进程重新召唤回前台，只需要使用 fg 命令即可。要注意的是，如果将进程挂起，则进程会暂停运行。如果要让进程在后台运行，可以使用 bg 命令。

6．screen 命令

screen 提供了一种虚拟的多屏幕技术来实现程序的后台运行。screen 所创造的虚拟屏幕像一个功能完全的终端，可以长时间稳定地运行各种程序，这对于长时间远程登录的用户可能很有实用价值。其常用的命令有以下几种格式。

- screen -dmS name：创建一个名为 name 的虚拟屏幕。
- screen -list：列出系统所创建的所有虚拟屏幕。
- screen -r name：重新连接到名为 name 的虚拟屏幕。

10.4.2 进程的优先级

进程的优先级决定了使用系统资源的顺序及响应速度，所以为了适应不同的需要，可以随时调整进程的优先级。

1．nice

nice 是调整系统优先级的命令，其使用格式如下：

```
nice [-n num] [process [参数]
```

命令的功能是将指定进程的优先级加上 num，然后执行。优先级用数字来表示，范围从-20（最高优先级）到 19（最低优先级）。num 可以为正数也可以为负数。如果觉得 FireFox 速度慢，也可以这样：

```
nice -n -10 FireFox
```

这条命令将 FireFox 的优先级提高 10 级（从数值上来看，是将优先级减去 10），然后运行 FireFox。注意无论是加还是减，优先级值均不能超出-20 到 19 的范围。

2．renice

nice 可以在运行前指定要运行的程序的优先级，但如果进程已经在运行，nice 就无能为力了。这时就可以使用 renice 命令来指定已经运行的程序的优先级。普通用户只能调整属于自己的进程的优先级，而 root 用户可以指定所有进程的优先级。renice 的格式如下：

```
renice num [参数]
```

命令的功能是将优先级加上 num（num 可正可负），然后将该优先级赋予参数指定的进程。

- -p n1 n2：重新设置 PID 为 n1、n2 的进程的优先级。
- -g gro：重新设置组名为 gro 的所有进程的优先级。
- -u user：重新设置进程所有者为 user 的所有进程的优先级。

将 PID 为 2305 及 10623 的进程，以及进程所有者为 westnorth 的所有进程优先级加 1 的命令如下所示。

```
renice +1  -u westnorth -p 2305 10623
```

10.4.3　进程自动调度 at、crontab

进程的自动调度在系统管理中使用得很频繁。所谓的自动调度，是指将指定的进程在特定的时间里自动运行的一种方式。而让进程自动运行，主要的方法是通过使用 at、cron、crontab 等命令。

1．at 命令

at 的功能是在某个指定的时间运行特定的命令。使用方式如下：

```
at [参数] TIME
```

部分参数的意义如下：

- -q：使用指定的队列存储自动运行命令。队列使用大写与小写各 26 个英文字母表示。
- -f file：从文件“file”中读入已经存在的命令，file 可以是自己建立的 at 可读取的命令集。
- -l：列出当前所有的自动运行命令。
- -d：删除指定的自动运行命令。
- -v：列出已经执行过且尚未删除的自动运行命令。

对于 at 命令来说，最重要的内容是 TIME 的设置。TIME 可以使用很多文字化的参数，如 today、weeks 等，其主要可用内容如下。

- 如果指定当天的时间，可用 hh：mm 来指定，hh 为小时，mm 为分钟，也可以使用 am、pm、midnight、noon、teatime 等单词来指定时间。
- 要指定不是当天的时间，可以用 MMDDYY 或 MM/DD/YY 的格式，其中 MM 是月份，DD 是日，YY 是年。
- 可以使用“now + 间隔”指定从现在起间隔一定时间段后的某个时间，其中的间隔可以是 minutes、hours、days、weeks。
- 可以使用 today 或 tomorrow 来表示今天或明天。

输入“at 时间”后，按下 Enter 键，会进入到一个交互模式来输入要在该时刻运行的命令。输入完命令后，按 Ctrl+D 键结束操作。下面是几个例子（输入下面的命令后，系统会提示要求用户输入要运行的命令，按照自己的需要输入相应的命令即可）。

```
at 4pm + 3 days      #三天后的下午 4 点运行特定的命令
at 10am Jul 31       #7 月 31 日上午 10 点运行命令
at 1am tomorrow.     #明天上午 1 点运行命令
```

2．自动周期运行命令 crontab

at 命令在指定的时间里只运行一次程序，如果要周期性地运行程序，应该使用 crontab 命令。其格式是：

```
crontab [参数]
```

参数表如下所示。

- -u user：设置用户 user 的日程表。
- -e：使用文本编辑器 nano 来编辑日程表。
- -r：删除系统中的日程表。
- -l：列出系统中已建立的日程表。

用法示例如下所示。

$crontab -l：列出用户目前的 crontab。

$crontab -e：进入 crontab 的编辑模式，编辑当前用户的日程表安排信息。进入编辑模式后，会看到有六列信息，这对于 crontab 来说是最为重要的内容。其指定了日程的执行周期与执行的命令。对于 crontab 命令的理解，最重要的内容是理解日程表的格式。其一共有 6 列，格式如下：

```
mm  hh  day  month  week  prog
```

mm 表示分钟，hh 表示小时，day 表示一个月份中的第几日，month 表示月份，week 表示一个星期中的第几天，prog 表示要执行的程序。编辑日程表，可用 crontab -e。

- 如 mm 项为*时，则说明命令 prog 要每分钟执行一次，而 hh 项为*时，则说明命令 prog 要每小时执行一次。其他项为*时，意义与之类似。
- 如 hh 项为 a-b 时则表示程序从第 a 小时到第 b 小时内需要运行，其他依次类推。
- 如 hh 项为*/n 时，则表示每 n 小时执行一次，其他依次类推。
- 如 day 项为“a,b,c”，表示程序在一月的第 a 天、第 b 天、第 c 天执行，其他依次类推。

用法示例：

```
#每天 7 点的第 0、30 分钟执行一次 top -n 2 b <monitor.log
0,30 7 * * * top -n 2 b <monitor.log
#在 12 月内，每天的早上 6 点到 12 点中，每隔 20 分钟执行一次/usr/bin/backup:
*/20 6-12 * 12 * /usr/bin/backup
#每月每天的午夜 0 点 20 分,2 点 20 分，4 点 20 分....执行 echo"haha">>/tmp/cron_haha
20 0-23/2 * * * echo "haha">>/tmp/cron_haha
#每两个小时执行一次 echo "haha">>/tmp/cron_haha
0 */2 * * * echo "haha">>/tmp/cron_haha
#晚上 11 点到早上 8 点之间每两个小时执行一次 echo "haha">>/tmp/cron_haha
0 23-7/2,8 * * * echo "haha">>/tmp/cron_haha
#每个月的 4 号和每个周一到周三的早上 11 点执行一次 echo "haha" >>/tmp/cron_haha
0 11 4 * mon-wed echo "haha">>/tmp/cron_haha
#1 月份早上 4 点执行一次 echo "haha" >>/tmp/cron_haha
0 4 1 jan * echo "haha">>/tmp/cron_haha
```

10.5 小结

本章从进程的基本概念开始，介绍了进程管理的基本知识，包括显示进程，以及终止进程等，在 10.4 节介绍了进程的优先级与进程的自动调度方法。由于进程管理在 Linux 日常使用中占据着很重要的位置，所以本章的内容应该重点掌握。动态进程监视程序 top 的使用与进程自动调度是本章内容中较难以掌握的，应参照书中的例子多加练习。

第 三 篇

Linux 应用篇

- 第 11 章　网络配置管理
- 第 12 章　网络相关软件
- 第 13 章　多媒体软件
- 第 14 章　图形图像软件
- 第 15 章　办公软件
- 第 16 章　专业排版软件 LaTeX

第 11 章　网络配置管理

Linux 的诞生和壮大与网络有着密切的关系，而且长期以来，Linux 被广泛运用于服务器平台，所以其网络功能极其强大。但要充分发挥其性能优势，以及解决在使用中的问题，还需要了解网络管理的基本知识。本章将介绍一些基本的网络管理知识。

11.1　网络管理概述

在 Ubuntu 的安装过程中，用户就已经开始了网络管理工作。如果选择文本界面的安装模式，Ubuntu 就允许用户自行设置 TCP/IP 网络的各项参数，如本机的 IP 地址、网关地址、DNS 地址等。如果选择图形界面的安装方式，Ubuntu 会自动设置这些参数（这也是对用户的区别对待，一般选择文本界面的用户均为有经验的用户，具备设置 TCP/IP 的能力）。通过这些参数，Ubuntu 就可以将本机注册到网络中。

要使用网络，首先要将计算机连接到网络。一般情况下，将电脑连接到网络使用的是一种称为网卡的设备。要使得网络连接可用，就需要网卡被系统正确识别。Linux 识别、管理网卡的方法与其他系统一样，均是通过加载该网卡的驱动程序来实现的。对 Ubuntu 而言，已加载的网卡驱动程序位于/lib/modules/2.6.24-16-generic/ubuntu/net（Ubuntu 8.04 版）。使用 tree 命令查看其结构，输出如下所示：

```
westnorth@11:06 /lib/modules/2.6.24-16-generic/ubuntu/net$ tree
.
|-- atl2
|   '-- atl2.ko
|-- et131x
|   '-- et131x.ko
'-- igb
     '-- igb.ko
     …
```

上述输出显示的内容包括系统安装的通用驱动程序，也包括用户曾经安装过的驱动程序。从输出中可以看出系统中已经投入使用网卡的驱动程序有三个，分别是 atl2.ko、et131x.ko、lgb.ko。

而 Ubuntu 支持的网卡列表在/lib/modules/2.6.24-16-generic/kernel/drivers/net。用 ls 命令查看，将其部分内容列出如下：

```
3c501.ko     cs89x0.ko     ibmlana.ko     plip.ko     smc-ultra.ko
…
acenic.ko    eexpress.ko   ne2.ko     sc92031.ko      veth.ko
bsd_comp.ko  hp100.ko      pcmcia     smc9194.ko
cassini.ko   hp.ko     pcnet32.ko     smc-mca.ko
```

```
...
chelsio      hp-plus.ko    phy          smc-ultra32.ko
```

从表中可以看到，Ubuntu 支持的网卡类型很多，基本涵盖了现在常用的所有网卡芯片。如果检测到网卡的存在，系统就会自动从驱动程序库中加载相应的驱动程序。如果系统正确安装了网卡，就会使用系统中的一些网络配置文件进行网络相关的设置。这些配置文件及其功能包括如下：

1．域名映射文件/etc/hosts

该文件的作用是将 IP 地址映射到主机名。用 vi 或任何文本编辑器打开该文件，可以看到其内容类似这样：

```
127.0.0.1 localhost
#将 127.0.0.1 映射到 localhost 主机名，需要 IP 地址时，输入 127.0.0.1 与输入 localhost 作用相同
192.168.2.22 mylap
192.168.2.2 www.earthfans.com westnorthweb
#这是完整格式的使用示例，作用是将 192.168.2.2 映射到 www.earthfans.com，同时给该网站取一个名为 westnorthweb 的别名。需要时，使用这三者意义均相同
```

2．网络设备界面配置文件/etc/network/interface

该文件是主要的网络设备界面文件，定义了各种网络设备的加载设置等。系统启动时由 ifup 使用该文件设置网络。该文件部分内容如下：

```
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.2.22
netmask 255.255.255.0
network 192.168.2.0
broadcast 192.168.2.255
gateway 192.168.2.2
# dns-* options are implemented by the resolvconf package, if installed
dns-nameservers 192.168.2.2
```

文件的内容意义如下。

- auto eth0：说明 eth0 会在启动时被自动加载。前面有 auto 标记的网络设备均会在系统启动时被 ifup -a 命令加载。
- iface eth0 inet static：定义了网络设备的使用方式，inet 说明 eth0 使用 TCP/IP 协议。static 说明网络采用静态 IP 分配方案。
- address 192.168.2.22：前面定义的网络设备（eth0）的 IP 地址。
- netmask 255.255.255.0：eth0 的掩码。
- network 192.168.2.0：C 类 IP 地址值。
- broadcast 192.168.2.255：广播地址。
- gateway 192.168.2.2：网关地址。
- dns-nameservers 192.168.2.2：DNS 服务器地址。

上述配置文件是使用静态 IP 地址，意思是网卡的 IP 地址等信息是固定的。如果是自动配

置 IP（DHCP 模式，该模式中，IP 地址、网关等信息均由 DHCP 服务器提供，无须用户设置），则在 interface 中添加如下一行即可：

```
    iface eth0 inet dhcp
```

3．启动脚本文件

系统启动时执行的初始化网络的脚本文件位于/etc/init.d/networking。该脚本文件有四个参数，分别是 start、stop、restart、force-reload。启动时使用的参数是 start，脚本文件中关于 start 的部分内容如下：

```
    start)
    log_action_begin_msg "Configuring network interfaces"
           type usplash_write >/dev/null 2>/dev/null && usplash_write "TIMEOUT
120" || true
    if [ "$VERBOSE" != no ]; then
        if ifup -a; then
         log_action_end_msg $?
        else
         log_action_end_msg $?
        fi
    else
        if ifup -a >/dev/null 2>&1; then
         log_action_end_msg $?
        else
         log_action_end_msg $?
        fi
    fi
           type usplash_write >/dev/null 2>/dev/null && usplash_write "TIMEOUT
15" || true
    ;;
```

从上述文件中可以看到，其最主要的工作是执行 ifup -a 命令，而 ifup -a 命令将按照网络设备界面配置文件/etc/network/interface 所设置的内容加载各项。

4．DNS 解析配置文件/etc/resolv.conf

这是 DNS 解析的关键配置文件，其内容如下所示：

```
    search launchmodem.com
    nameserver 61.178.0.93
    nameserver 202.100.64.68
```

- nameserver：设置 DNS 服务器的 IP 地址，这是该文件最重要也是不可缺少的选项。
- search：设定域名的搜索列表，可以指定多个地址。

11.2 iptables 简介

iptables 顾名思义是 IP（Internet Protoco，互联网协议）数据包的规则表。使用 iptables 可以建立强大且灵活的防火墙。

1．iptables 的功能

当电脑连接到网络时，为了使用特定的功能，会开放一些通道，通过这些通道与网络中其他计算机交流，这些通道称为端口。如果向网络中其他计算机提供 WWW 服务，就需要开放 80 端口供其他用户访问；如果要获得 DNS 服务，就需要接收来自其他计算机的 53 端口的数据。不论是其他计算机通过某些端口访问本机资源，还是接收其他计算机机的数据，这些数据均是通过 IP 包来进行传输的。iptables 规定了这些包传输的规则，如哪些包可以通过、哪些包被拒绝，以及本机的哪些端口是开放的等。

2．规则表与规则链

iptable 是一个很强大但也很复杂的命令，可以用来设置 IP 包的 table（规则表）。每个规则表里可以有许多规则链（chain），每个规则链匹配一种 IP 包，规则链匹配的包称为目标包。目标包有如下四种操作。

- ACCEPT：是指接受该包，让包通过。
- DROP：是指简单地将包丢弃。
- QUEUE：是指将包加入用户队列。
- RETURN：表示不再让包继续匹配本规则表中其余规则链，而是启动下一条规则表。

系统内置了四种规则表，分别是 filter、nat、mangle 和 raw。下面简单介绍一下最常用的 filter 与 nat 规则表。

- filter：这是默认规则表，包含三种内建的规则链，INPUT（接收从网络中其他计算机发来的包）、FORWARD（转发从网络中其他计算机发来的包）和 OUTPUT（向网络发送的包）。
- nat：本规则表在建立新连接时被使用，作用是对包进行默认转换。包含三种内建的规则链：PREROUTING（包进入时转换）、OUTPUT（包进入路由前转换）、POSTROUTING（包发送前转换）。

3．使用 iptablse 建立静态防火墙

下面我们以建立一个静态防火墙为例说明其作用，具体步骤如下（均为在终端中输入的命令）。

（1）除去规则表中所有规则链，其命令如下：

```
iptables -F
```

-F 参数的意义是除去规则表中规则链。其后若加上名称，则除去指定的规则链。若不加规则链名称，则除去表中所有的规则链。这么做的原因是预防新建规则与其他规则冲突。

（2）删除规则链，命令如下：

```
iptables -X
```

-X 参数的意义是删除规则链，如果没有显式指定任何链名，则删除所有的规则链。此命令只用于删除空规则链，所以应在-F 参数后使用。

（3）清除 nat 规则表中的规则链，命令如下：

```
iptables -t nat -F
```

-t 参数指定规则表，这里指定的规则表是 nat。

（4）删除 nat 规则表中的规则链，命令如下：

```
iptables -t nat -X
```

删除 nat 规则表中的规则链。

（5）将所有 INPUT 链匹配的包全部丢弃，命令如下：

```
iptables -P INPUT DROP
```

-P 参数的意义是设置特定链的规则。本例命令的含义是匹配 INPUT 链，并将匹配的链全部丢弃（DROP）。

（6）添加 INPUT 链关于网卡的处理规则，其命令如下：

```
iptables -A INPUT -i ! eth0 -j ACCEPT
```

- -A：给指定的链添加处理规则。
- -i：将规则插入到指定的链。本例的规则匹配不是来自于网络接口 eth0 的包。
- -j：指定匹配链后的操作。本例是接受匹配的包。

这条规则的意思是：匹配所有不是来自于网络接口 eth0 的数据包，并接受这些包。

（7）添加 INPUT 链关于协议与端口的处理规则，其命令如下：

```
iptables -A INPUT -i ppp0 -p tcp --sport 80 -j ACCEPT
```

- -p：指定包所使用的协议，可以取的值包括 tcp、udp、icmp、all。
- --sport：指定源端口，也就是包来源的端口。同时该参数可以取一个端口范围，如 100：200。

这条规则的意思是：允许来自网络接口 ppp0，采用 TCP 协议，并且来源端口是 80 的数据包进入。

（8）添加 INPUT 链关于协议、端口的处理规则。命令如下：

```
iptables -A INPUT -i ppp0 -p udp --sport 53 -j ACCEPT
```

这条规则的意思是：允许来自网络接口 ppp0，采用 UDP 协议，并且来源端口是 53 的数据包进入。端口 53 是 DNS 端口。

（9）查看链中的规则。命令如下：

```
iptables -L -v
```

- -L：列出当前链中所有的规则。
- -v：列出详细信息。

（10）添加 INPUT 链关于协议、端口的处理规则。命令如下：

```
iptables -A INPUT -i ppp0 -p tcp --dport 80 -j ACCEPT
```

--dport：指定目的端口，本例指定开放本机的 80 端口。80 端口提供的服务是 WWW，这样其他用户就可以访问本机的 WWW 服务了。

这样就建立了一个初步的防火墙。在建立过程中，先是拒绝所有的 INPUT，然后在 6、7、9、10 步分别向链中插入允许 INPUT 的规则。这样，就只允许这四个规则匹配的链。这种技术，在网络管理中一般称为白名单，意思是先禁止所有的连接，然后列出一个允许连接的名单，只在名单上的连接才会被允许。与之相反的被称为黑名单，相关内容在本书 21.2 节已进行了介绍，有兴趣的读者可自行查阅。

iptables 在 Linux 网络管理中有着极其重要的作用。在 Ubuntu 中，iptable 几乎涵盖了所有的网络管理范围，如端口转发、端口映射、透明代理等工作。由于该内容很难理解与掌握，这

里就不再详细介绍了，有这方面需求的读者可查阅 iptables 的手册。

11.3　监视网络性能

虽然 Linux 是一款稳定强大的操作系统，在网络操作中具有很大的优点，但是由于网络的结构复杂，在实际使用中，网络还可能会出现许多的问题，如性能下降、网络通信中断等。这时，就需要可以动态监视网络性能的软件，让用户可以随时了解网络的状态，方便解决各种问题。本节将介绍相关的网络监视命令。

11.3.1　使用 ping 检测网络性能

ping 命令是网络中断或网速过慢时大多数用户会首先使用的命令，该命令可以查询与指定网络地址的通信状况。ping 可接受 IP 地址或域名作为查询参数，其最简单的使用方法如下：

```
ping localhost
```

这条命令测试与 localhost 的通信状况，localhost 是本机地址。当然 localhost 也可以替换为任何格式正确的 IP 地址（如 192.168.2.22）或任何存在的域名（如 www.google.com）。如果与该地址是连通的，则输出可能类似于这样：

```
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.046 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.043 ms
--- localhost ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5003ms
rtt min/avg/max/mdev = 0.043/0.044/0.046/0.006 ms
```

第一行“PING localhost (127.0.0.1) 56(84) bytes of data”，说明要查询的目标地址是 localhost，ping 将域名转换为 IP 地址“127.0.0.1”，然后发送 ICMP 报文的 ECHO_REQUES 请求到该地址。

第二、三行“64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.046 ms”，意思是将 64B 的数据发送到 localhost（64B 是 ICMP 报文的长度）。icmp_seq 是发送 ICMP 报文的顺序号、time 是从发送 ICMP 报文到接收到回应信息使用的时间长度。time 的值是反映网络速度的最直观的一个数据，如果 time 的数值大于 400ms，则可认为网络速度过于缓慢。如果出现“Destination Host Unreachable”这样的字样，则说明两台机器之间的网络连通有问题。

第四行“--- localhost ping statistics ---”意思是下面的内容均是统计信息。

第五行“6 packets transmitted, 6 received, 0% packet loss, time 5003ms”意思是发送了 6 个包，接收了 6 个回应信息，0%的包丢失，共花费了 5003ms。

第六行“rtt min/avg/max/mdev = 0.043/0.044/0.046/0.006 ms”，显示返回时间，分别是最小值、最大值等。

ping 命令还有许多其他参数，常用的一些参数罗列于下。

- -b：允许 ping 广播地址。
- -c count：指定发送报文的数量，发送报文数量达到 count 后，就停止发送。
- -f：洪水 ping，对于发送的每个 ECHO_REQUEST 均显示一个“ . ”，而如果接收到一

个“ECHO_REPLY”，则将输出一个退格键，删除前面显示的“ . ”，就可以清楚地看到网络发送和接收报文的情况。同时，这也是一种网络攻击手段，称为 ICMP 攻击。

- -l preload：预加载包。不用等待目标机器返回信息就发送 preload 个数据包，默认方式是发送包后等待返回然后再发送下一个包。普通用户的 preload 只能小于 3。
- -i interval：指定发送报文的时间间隔。默认间隔是 1s，而如果使用洪水 ping 方式则是无间隔，所以可以想象，如果许多计算机向同一台服务器发送洪水 ping 会造成多么严重的网络阻塞！不过现在大多数服务器均配置了防火墙，可以有效地阻止洪水 ping。“interval”是时间间隔，以秒为单位。普通用户只能指定大于 0.2s 的间隔。
- -n：输出时不显示域名，只显示 IP 地址。
- -p pattern：用“pattern”填充发送的报文，如 ping -p ff。这在诊断数据内容相关的错误时很有用。
- -R：记录路由信息。
- -s size：指定报文大小，默认是 64。
- -t：设置 IP 包的生存时间（TTL）。
- -w：设置发送报文的总时间，单位是 s，超过该时间，ping 命令会自动退出。
- -W：设置等待回应的时间，如果超出该时间还没有任何回应，则 ping 退出。



在检测与其他网络通信是否畅通前，应该先使用 ping 命令检查与 localhost 的连接情况。如果与 localhost 连接正常，说明本机的网络设置、网卡驱动均正常。在本机网络设置正确的情况下，再检查与外部网络的连接情况，可以减少误判的情况发生。

11.3.2 使用 netstat 监视网络性能

netstat 用来显示网络相关的信息状态，如网络连接情况、路由表、接口统计信息、伪装连接、多播成员等。netstat 的使用格式如下：

```
netstat 参数 1 参数 2
```

netstat 显示的内容由参数 1 指定，参数 1 的可取值如下。

- （空）：如果第一个参数为空，则显示打开的套接字。
- --route，-r：显示核心路由信息。
- --groups，-g：显示多播组成员信息。
- --interfaces，-i：显示所有网络接口。
- --statistics，-s：显示各种协议的统计信息。

参数 2 的常用值列表如下所示。

- --protocol=family、-A：指定使用的协议，可以使用 inet、unix、ipx、ax25、netrom，以及 DDP 等协议。
- -c、--continuous：持续每秒显示一次相关信息。
- -e、--extend：显示附加信息，使用本参数两次可以显示全部细节。
- -o、--timers：显示网络时间相关的信息。
- -p、--program：显示套接字的属主程序的 PID 号。
- -l、--listening：仅显示监听的套接字。

- --numeric、-n：显示数值形式的域名、端口、用户名等信息。
- -a、--all：显示所有套装字，包括监听与非监听的。
- -F：显示从 FIB 得到的路由信息。
- -C：显示从路由缓冲得到的路由信息。

下面举几个例子来介绍命令的使用方法，以及其输出内容的意义。

1．显示所有套接字

使用-a、-n 参数可以显示出所有套接字的状态信息，并且以数值形式显示所有信息。使用的命令及输出如下所示：

```
Awestnorth@22:51 /etc$ netstat -a -n
Active Internet connections (servers and established)#以下是网络连接状态
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0   127.0.0.1:631          0.0.0.0:*               LISTEN
#上一行说明本机的 631 端口处于监听状态，并且使用 TCP 协议
tcp       0    10     192.168.1.200:51558   192.168.1.101:5900  ESTABLISHED
#上一行输出说明本机与 192.168.1.101 建立了连接。本机使用的端口是 51558，而远程机使用的端口是 5900，下面的三行输出内容与前两行相似，读者可自行分析
tcp6       0      0   :::5900               :::*                    LISTEN
udp        0      0   0.0.0.0:5353          0.0.0.0:*
udp        0      0   0.0.0.0:46442         0.0.0.0:*
Active UNIX domain sockets (servers and established)#以下是 UNIX 域套接字信息
Proto RefCnt Flags     Type       State     I-Node   Path
unix  2      [ ]       DGRAM                  6282     @/com/ubuntu/upstart
unix  6      [ ]       DGRAM                  12869    /dev/log
unix  2      [ ]       DGRAM                  6432   @/org/kernel/udev/udevd
```

2．显示内核路由信息

内核路由信息是指由本机的网卡连接到网络其他主机的路由信息，显示内核路由信息的命令如下：

```
netstat -nr
```

输出如下：

```
Kernel IP routing table
Destination     Gateway         Genmask     Flags Metric   Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 wlan0
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 wlan0
```

这些输出的意义将在本书 11.6 节路由管理中进行介绍，这里就不加说明了。

3．显示使用 TCP、UDP 协议的传输状态

netstat 还可以显示使用特定协议的传输状态。如使用 t 参数显示使用 TCP 协议的传输状态，而使用参数 u 则显示使用 UDP 协议的传输状态。示例如下：

```
earthfans@08:20 ~$ netstat -t          #显示使用 TCP 协议的连接
#以下是输出
激活 Internet 连接 (w/o 服务器)
```

```
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp     0      0 mylaptop.local:37203    220.181.6.18:www       ESTABLISHED
#本机的端口 37203 连接到 220.181.6.18，连接已经建立。
tcp     0      0 mylaptop.local:36863    ti-in-f18.google.:https ESTABLISHED
tcp     0      0 mylaptop.local:57137    61.172.250.107:www     TIME_WAIT
#本机的端口 57137 连接到 61.172.250，107，连接正在建立过程中。
tcp     0      0 mylaptop.local:36870    ti-in-f18.google.:https ESTABLISHED
…
earthfans@08:20 ～$netstat -u    #显示使用 UDP 协议的连接
#由于普通用户使用 UDP 服务的连接比较少，所以输出可能是空。不过该命令的输出与使用参数 t 时格式是一样的，可以参照上述内容进行理解
```

11.4 配置 TCP/IP

在安装 Linux 时，可能已经配置了 TCP/IP 的参数。但在实际使用过程中，会发生许多需要更改这些参数的情况，如计算机所分配的 IP 地址发生了变化、需要采用动态分配 IP 的方式、需要更改路由信息等。这些操作可以通过 ifconifg 命令来实现。

ifconfig 命令如果后面不加任何参数，就会显示当前活动的网络设备的相关信息，如下所示：

```
$ifconfig                  #后面未加参数，显示当前活动的网络设备的 IP 地址、MTU 等信息
eth0      Link encap:以太网  硬件地址 00:03:0d:5e:e3:8f
          inet 地址:192.168.2.22  广播:192.168.2.255  掩码:255.255.255.0
          UP BROADCAST MULTIChAST DYNAMIC  MTU:1500  跃点数:1
          接收数据包:0 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:0 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:0 (0.0 B)  发送字节:0 (0.0 B)
          中断:17 基本地址:0x2000
lo        Link encap:本地环回
          inet 地址:127.0.0.1  掩码:255.0.0.0
          inet6 地址: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  跃点数:1
          接收数据包:77 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:77 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:0
          接收字节:7055 (6.8 KB)  发送字节:7055 (6.8 KB)
```

如果加参数-a，就会显示系统所有可用的设备（包括已被关闭的设备）。但 ifconfig 的功能远不止于此，其不仅可显示网络设备的相关信息，还可完成很多网络设置工作。ifconfig 完整的使用格式如下：

```
ifconfig [-v] [-a] [-s ] [网络设备界面]   或
ifconfig 网络设备界面 选项 | 地址 ...
```

网络设备界面参数主要是指网络设备，如网卡，也包括一些协议，如 PPPoe 等。“选项”是完成各项工作的主要部分，一些最常用的选项列举如下。

- -v：显示更详细的信息。
- -s：显示短格式的信息。
- up：激活一个设备，该设备应该是绝对可以信任的设备。
- down：关闭一个激活的设备。
- mtu 值：设备最大传输单元（MTU）的大小。
- netmask 地址：设置掩码地址。
- [-]broadcast [地址]：设置广播地址。
- address：设置 IP 地址。
- hw 种类地址：设置硬件地址。

下面讲解几个常用的 TCP/IP 配置操作。

11.4.1　显示特定网络设备的状态

ifconfig 可以指定要显示的网络设备。也就是说，可以只让 ifconfig 输出指定设备的信息。只让 ifconfig 显示网卡 eth0 的相关信息，而不显示其他设备的示例如下：

```
earthfans@21:26 ~$ ifconfig eth0            #该命令只显示 eth0 的详细信息
eth0      Link encap:以太网  硬件地址 00:03:0d:5e:e3:8f
          UP BROADCAST MULTICAST  MTU:1500  跃点数:1
          接收数据包:0 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:0 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:0 (0.0 B)  发送字节:0 (0.0 B)
          中断:16 基本地址:0x2000
```

11.4.2　设置网络设备的 TCP/IP 参数

ifconfig 可以设置网络设备的 TCP/IP 参数，包括 IP 地址、子网掩码等，一些设置的示例如下：

```
sudo ifconfig eth0 192.168.2.2 netmask 255.255.255.0   #设置 eth0 的 IP 地址
                                                          和子网掩码值。
sudo ifconfig eth0 hw ether 00:03:0d:5e:e3:8f       #设置 eth0 的 mac 地址。
sudo ifconfig dynamic                      #设置当前设备 TCP/IP 地址为 dhcp 模式
sudo ifconfig eth0 down
#关闭网卡，如果使用 up 选项则为启动网卡
sudo iwconfig wlan0 essid "west"
#使无线网卡 wlan0 连接到名为 west 的无线网中
sudo iwconfig wlan0 channel auto          #设置无线网卡 wlan0 自动选择连接频道
sudo route add default gw 192.168.1.1   #使 192.168.1.1 成为默认网关
```

11.4.3　给网络设备取别名

ifconfig 还可以给网络设备（如网卡）取一个别名，这样就可以给一个网卡同时设置几个不同的 IP 地址，而且几个 IP 地址可以属于不同的网段。这样在同一个机器要使用不同的网段时比较有用。

设网卡 eth0 原 IP 地址是 192.168.2.22，使用下列命令给其设置一个名为 eth0:1 的别名：

```
ifconfig eth0:1 192.168.155.2 netmask 255.255.255.0
```

设置了网卡的别名后，再用“ifconfig -a”查看，输出如下：

```
eth0      Link encap:以太网  硬件地址 00:03:0d:5e:e3:8f
          inet 地址:192.168.2.22  广播:192.168.2.255  掩码:255.255.255.0
          UP BROADCAST MULTICAST DYNAMIC  MTU:1500  跃点数:1
          接收数据包:0 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:0 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:0 (0.0 B)  发送字节:0 (0.0 B)
          中断:17 基本地址:0x2000
eth0:1    Link encap:以太网  硬件地址 00:03:0d:5e:e3:8f
          inet 地址:192.168.155.2  广播:192.168.155.255  掩码:255.255.255.0
          UP BROADCAST MULTICAST DYNAMIC  MTU:1500  跃点数:1
          中断:17 基本地址:0x2000
lo        Link encap:本地环回
          inet 地址:127.0.0.1  掩码:255.0.0.0
          inet6 地址: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  跃点数:1
          接收数据包:77 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:77 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:0
          接收字节:7055 (6.8 KB)  发送字节:7055 (6.8 KB)
```

可以看到，网络设备有三个：eth0，eth0:1，lo。eth0 是物理网卡，eth0:1 是该物理网卡的一个别名。而且 eth0 的 IP 地址是 192.168.2.22，eth0:1 的 IP 地址是 192.168.155.2，分属于不同的网段。

11.5 DNS 管理简介

DNS 服务就是将用户输入的类似 www.google.com 的域名转换为 IP 地址值 64.233.189.147 的过程。本节主要介绍在本机维护 DNS 服务的方法，以及与 DNS 相关的几个命令的用法。

11.5.1 DNS 简介

网络中的资源是通过一台台称为“服务器”的计算机来提供的。为了使用户更方便地访问这些资源，一个专门的机构给这些服务器分配了各种不同的地址。而为了使路由器等网络设备能更快捷准确的管理，这些地址被严格限制为四组均小于 255 的数字，如 192.168.1.22、202.100.0.12，这些地址被称为 IP 地址。IP 地址虽然对于网络设备来说容易管理，但是对于人类来说就显得有点不太“友好”了。这是由于人类更容易记住一些有意义的字符，而很难记住这些无意义的数字地址。

为了解决这个问题，互联网管理机构又在网络中增设了一种专门的服务器用来将这些 IP 地址转换为人类友好的字符地址，如将 64.233.189.147 转换为 www.google.com。简单来说，这些服务器不过是维护了一张表，表中的内容一列是 IP 地址，另一列对应人类可识别的字符地址。

当用户输入文字地址，如 www.google.com，然后发送时，这些地址信息被送到 DNS 服务器，服务器在表格中查询 www.google.com，得到该地址对应的 IP 地址，如 63.233.189.147，然

后重定向到实际 IP 地址，就完成了 DNS 转换。为了最大可能地提高性能，世界各地有许多的 DNS 服务器，这样在用户在需要时，总可以就近得到最快速的 DNS 服务。

11.5.2　DNS 的配置文件

对于 Ubuntu 而言，其也在系统内部维护了一个 DNS 服务器，这样就可以不用经过外部 DNS 服务器就完成域名到 IP 地址的转换了。这样做不但可以提高网络浏览的性能，对于网络其他应用也有很重要的作用。如在网络应用、管理中经常使用的 localhost，就是使用了系统 DNS 服务完成转换的，其实际 IP 地址是 127.0.0.1。下面介绍几个重要的 DNS 相关配置文件。

1．/etc/host.conf

该文件指定如何解析主机名，其内容如下：

```
# The "order" line is only used by old versions of the C library.
order hosts,bind
multi on
```

- order 指定了查询顺序。本例的意思是先使用/etc/hosts 文件查询，然后使用 DNS 服务。
- multi 指示如何处理查询/etc/hosts 获得的多个 IP，on 代表返回多个 IP（也就是说主机可以有多个 IP 地址），off 代表就算获得了多个 IP，也只返回第一个 IP。

详细内容可以查阅 host.conf 的手册，命令是 man host.conf。

2．/etc/resolv.conf

该文件是 DNS 服务的主要配置文件。可以设置本地域名、设置 DNS 服务器地址等信息。系统中/etc/resolv.conf 文件的内容可能类似于这样：

```
# Generated by NetworkManager
nameserver 192.168.1.1
#domain earthfans
#search
```

- nameserver 除指定 DNS 服务器的 IP 地址外，还可以使用下列指令。
- domain：指定本地域名。
- search：指定 DNS 服务器列表。

3．/etc/hostname

该文件的主要作用是设置系统主机名。对于 Ubuntu 系统而言，这个文件中只有一行内容，就是本机在网络上显示的主机名。

4．/etc/hosts

设置一个简单的 DNS 表格，可以处理本机简单的 DNS 服务。处理过程是收到主机名，然后将主机转换为相应的 IP 地址返回。系统中的 hosts 文件应该类似于这样：

```
# ip 地址     主机名                  别名
127.0.0.1       localhost
127.0.1.1       westnorth-laptop
```

在这里，就可以看到经常使用的 localhost，其实际 IP 地址是 127.0.0.1（也就是本机）。当然，也可以将其他地址加入。如果在本地网中还有一台计算机，其 IP 是 192.168.1.101，一个

IPOD 的 IP 地址是 192.168.1.123，则可以将下面内容加入/etc/hosts 中，以便进行管理。

```
# IP 地址      主机名                  别名
192.168.1.101   westnorth-desktop       pc1
192.168.1.123   westnorth-ipod          ipod
```

将这些文件加入 hosts 后，就可以方便地使用其主机名或别名来检查连接情况了。下面是分别使用别名及主机名进行查询的示例。

在 ping 中使用别名进行查询。查询 IP 地址为 192.168.1.123 的主机的示例如下，在命令中并没有使用 IP 地址，而是使用了别名 ipod 进行查询。

```
westnorth@06:27 /etc$ ping ipod
#使用别名 ping,实际上 ping 的是 IP 地址为 192.168.1.123 的主机。这是在文件/etc/hosts 中设置的
PING westnorth-ipod (192.168.1.123) 56(84) bytes of data.
64 bytes from westnorth-ipod (192.168.1.123): icmp_seq=1 ttl=64 time=222 ms
…
--- westnorth-ipod ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3013ms
rtt min/avg/max/mdev = 2.784/90.268/222.773/95.285 ms
```

同样地，在 ping 中也可使用主机名称进行查询。这也是最常见的查询方式，如 ping www.google.com 就使用了主机名进行查询。在/etc/hosts 中定义自定义的主机名 westnorth-desktop 的例子如下：

```
westnorth@06:27 /etc$ ping westnorth-desktop            #使用主机名 ping
PING westnorth-desktop (192.168.1.101) 56(84) bytes of data.  #实际的 IP 地址是 192.168.1.101
64 bytes from westnorth-desktop (192.168.1.101): icmp_seq=1 ttl=64 time=4.94 ms
--- westnorth-desktop ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.377/3.158/4.940/1.782 ms
```

由上例可以看出，将本地主机名与对应的 IP 地址加入 hosts 表后，就可以方便地使用各种方式（如别名、主机名）对这些主机进行操作了。

也可以将常用网站的 IP 地址加入其中，在输入网址后就由本机完成 DNS 转换，不用连接到外部 DNS 服务器上，这样可以提高网络性能。笔者将下列内容加入到/etc/hosts 中，感觉上网速有一定的提高，读者可根据自己的情况，加入自己经常使用的网站。

```
64.233.189.104 www.google.com  google
220.181.43.121 mp3.baidu.com mp3
59.36.101.19    forum.ubuntu.org.cn
211.147.3.93    www.douban.com  douban
203.208.39.104   www.google.cn  g
64.233.189.147   reader.google.com reader
222.73.207.160  www.verycd.com
```

11.5.3　nslookup 命令

nslookup 命令可以用来查询 DNS 服务器中域名与 IP 地址的映射关系。简单地说，就是查询一个网址其对应的 IP 地址。如果要查询单个域名，直接在 nslookup 后面跟一个域名即可。查询 www.google.com IP 地址的命令如下：

```
nslookup www.google.com
```

这样就可得到 www.google.com 的 IP 地址。如果要查询多个域名，可不带参数输入 nslookup 进入交互模式查询。在交互式查询模式下，可以多次输入不同的域名进行查询，还可以对查询的行为作出一些规定。交互式模式中可以使用的命令如下所示。

- lserver：查询指定的 DNS 服务器，并将其设置为默认的 DNS 服务器。
- set 命令：修改 nslookup 的一些参数。可以设置的参数如下：

 set class：设置查询的类型，一般情况下为 Internet(IN)，还可以是 Chaos(CH)、Hesied(HE)等。

 set debug：设置为调试模式。

 set nodebug：设置为非调试模式。

 set d2：设置为详细调试模式。

 set nod2：设置为非详细调试模式。

 set domain：设置默认域名。

 set port：重新设置 DNS 的端口，默认是 53。

 set q(querytype)：更改查询的类型，默认的类型为 A 记录。

 set retry：设置重试的次数，默认值为 3。

 set timeout：设置超时时间（单位为秒），如果大于该时间仍没有返回信息，则判断为超时。

如使用交互查询模式查询邮件交换信息的示例如下。

（1）不带参数运行 nslookup，进入交互查询模式：

```
westnorth@06:43 /etc$ nslookup      #不带参数运行，进入交互模式
```

（2）使用 set 命令修改参数，查询邮件交换信息：

```
> set querytype=MX                  #设置查询类型为邮件交换信息
```

（3）输入要查询邮件交换信息的网站名称，本例输入为 www.google.com。

```
> www.google.com                    #查询 www.google.com 的邮件服务信息
#下面的输出是邮件服务器的信息
Server:     192.168.1.1
Address:    192.168.1.1#53
Non-authoritative answer:
www.google.com  canonical name = www.l.google.com.
www.l.google.com    canonical name = www-china.l.google.com.
Authoritative answers can be found from:
l.google.com
origin = g.l.google.com
mail addr = dns-admin.google.com
serial = 1364450
```

```
refresh = 900
retry = 900
expire = 1800
minimum = 60
```

下面是一个查询 A 记录的示例。同样不带参数运行 nslookup 进入交互查询模式，然后参照如下步骤查询 www.google.com 的 A 记录。

（1）使用 set 命令修改参数为 A，查询 A 记录。

```
> set querytype=A          #设置查询类型为 A 记录，这是普通的域名至 IP 地址转换的功能
```

（2）输入要查询 A 记录的网站名称。

```
> www.google.com          #再查询 www.google.com，可以看到输出与刚才完全不同
#下面的输出是 A 记录的信息
Server:     192.168.1.1
Address:    192.168.1.1#53
Non-authoritative answer:
www.google.com  canonical name = www.l.google.com.
www.l.google.com    canonical name = www-china.l.google.com.
Name:   www-china.l.google.com
Address: 64.233.189.147
Name:   www-china.l.google.com
Address: 64.233.189.99
Name:   www-china.l.google.com
Address: 64.233.189.104
```

nslookup 还可设置其他查询类型，以获得特定的信息，这些类型如下所示。

- SOA：标记一个区的开始。
- A：主机对应的 IP 地址。
- MX：路由邮件交换器。
- NS：标识一个区的权威服务器。
- CNAME：主机的别名信息。
- TXT：文本记录。

11.5.4 dig 命令

dig 是一个与 nslookup 类似命令，可以获得服务器相关信息。该命令为非交互式，但是提供的信息要多于 nslookup 所提供的，其使用格式如下：

```
dig 服务器 主机名 记录类型 类别 选项
```

其中各选项的意义如下所示。

- 服务器：查询域名中用到的 DNS 服务器，与 nslookup 交互模式中的 server 命令功能相同。
- 主机名：要查询的域名。
- 记录类型：设置查询的记录类型，相当于 nslookup 中的 set querytype，默认值为 A。
- 类别：设置查询的类别，相当于 nslookup 中的 set class，默认值为 IN（Internet）。

- 选项：查询选项，可以指定另一个端口（默认的 DNS 端口为 53）等。

使用 dig 命令查看新浪网邮件交换器信息终端输出情况的例子如下：

```
westnorth@17:11 ~$ dig www.sina.com MT IN
#要查询的主机名是 www.sina.com，记录类型是 MT，而查询类别是 IN
; <<>> DiG 9.5.0-P2 <<>> www.sina.com MT IN
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36881
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;www.sina.com.          IN  A
;; ANSWER SECTION:
#下面的输出显示了新浪网络服务器的具体信息
www.sina.com.       15  IN  CNAME   us.sina.com.cn.
us.sina.com.cn.     22  IN  CNAME   news.sina.com.cn.
news.sina.com.cn.   17  IN  CNAME   jupiter.sina.com.cn.
jupiter.sina.com.cn.    51  IN  A   218.30.66.101
;; Query time: 30 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Sun Nov 30 17:12:00 2008
;; MSG SIZE  rcvd: 115
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17017
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0
;; QUESTION SECTION:
;MT.                IN  A
;; AUTHORITY SECTION:
MT.         10800   IN  SOA ns.MT.  tech.nic.org.MT.  2008102000  1800  900
604800 86400
;; Query time: 240 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Sun Nov 30 17:12:01 2008
;; MSG SIZE  rcvd: 72
```

11.6　路由管理简介

路由信息就是网络中信息传输所经过的服务器记录。如果发生网络故障，则应该先使用路由查询命令来检查网络传输的问题出在什么地方。下面将依次介绍查询路由的方法。

1．tracepath 命令

Ubuntu 中路由查询的命令是 tracepath。其使用方法很简单，可以使用域名或 IP 地址作为命令的参数。查询命令如下所示，使用其中任何一类语句即可。

```
tracepath www.google.com
tracepath 123.232.2.123
```

该命令的运行结果会显示源主机（发出查询命令的计算机）到目的主机之间的网关，同时还可显示发出的信息包在每个网关的往返时间。

2．route 命令

Ubuntu 中还可以使用 route 命令来查看与管理系统核心路由。使用参数 n 则显示核心路由信息，示例如下所示：

```
westnorth@19:52 ～$ route -n                #显示核心路由信息
Kernel IP routing table
Destination     Gateway         Genmask     Flags Metric   Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 wlan0
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 wlan0
```

由上面的输出可以看出，192.168.1.1 是网关，同时该网关是开启的（由 Flags 的“UG”状态判断，U 说明该设备是开启的，G 说明是网关）。

Route 命令的一些基本用法如下：

```
route add -net 127.0.0.0 netmask 255.0.0.0 dev lo
#添加标准环回接口路由
route add -net 192.56.76.0 netmask 255.255.255.0 dev eth0
#添加一个经由 eth0 到达本地网络 192.5.7.0 的路由，关键字 dev 可以省略
route del default
#删除默认路由
```

3．查看文件/proc/net/route

“cat /proc/net/route”的输出结果是以十六进制表示的核心路由表，与 route 的输出结果本质上是一致的，示例如下：

```
westnorth@19:56 ～$ cat /proc/net/route
#显示核心路由信息，显示为 16 进制信息，其含义与上一条命令相同
Iface   Destination    Gateway     Flags   RefCnt  Use    Metric  Mask    MTU
TU      Window  IRTT
wlan0   0001A8C0      00000000      0001    0       0       0       00FFFFFF 0
wlan0   00000000      0101A8C0      0003    0       0       0       00000000 0
```

输出内容中各项的含义如下所示。

- Destination：目的主机。
- Gateway：经过的网关。
- Genmask：目的主机的掩码。
- Flages：路由标志。
- Iface：该路由使用的网络设备。
- Metric：路由成本。
- Ref：该路由被使用的次数。

11.7　端口管理简介

在网络使用中，会经常接触到有关端口的操作，如有时为了使用某些服务，需要打开某些端口。而有时为了系统的安全起见，又需要关闭一些端口。因此，为对端口进行操作，需要了

解一些端口的基本知识。

1．etc/srevices 配置文件

/etc/services 文件是端口管理中重要的配置文件，其主要功能是设置某项网络服务名称与其对应的端口、协议等内容。其部分内容如下所示。

```
#名称        端口/协议        别名              注释
chargen      19/udp      ttytst source
FTP-data     20/tcp
FTP      21/tcp
fsp      21/udp      fspd
ssh      22/tcp              # SSH Remote Login Protocol
ssh      22/udp
telnet   23/tcp
smtp     25/tcp      mail
time     37/tcp      timserver
time     37/udp      timserver
rlp      39/udp      resource    # resource location
nameserver   42/tcp      name        # IEN 116
whois        43/tcp      nicname
```

由文件可以看出，FTP 服务使用的是 TCP 协议，占用了 21 端口。而 SSH 协议使用了 UDP 与 TCP 协议，占用了 22 端口。这些端口地址可以进行更改，如将 FTP 服务的端口修改为 2121，然后在使用 FTP 服务的程序中也进行相关的更改即可。修改服务的端口值可以在一定程度上提高网络的安全性能。

2．打开端口

在 Ubuntu 中，打开端口可以简单地使用 netcat 命令（nc 是 netcat 的缩写形式）。如要打开端口 10000，则可使用如下代码：

```
nc -lp 10000 &
```

3．端口映射

在一个局域网中，有时内网的计算机没有直接连接到外部网上，所以无法正常进行某些操作，如 bt 下载或 emule 下载。这时，就可以使用端口映射技术，将网关的端口映射到内部网中的某台计算机上。这样，对该计算机而言，就可像直接连接到外部网上一样使用映射的端口。

假设一台计算机通过 PPPOE（也就是 ADSL 拨号方式）连接到互联网，但同时其也充当某局域网的网关。而该局域网内有一台计算机（设其 IP 地址为 192.168.1.200）要使用 11197 端口做 BT 下载。这里就要在充当网关的计算机上设置端口映射。本例使用 iptables 来完成端口映射，其命令如下：

```
iptables -t nat -A PREROUTING -p tcp -i ppp0 --dport 11197 -j DNAT
--to-destination 192.168.1.200:11197
```

11.8　远程登录简介

在实际的工作与生活中，经常会遇到需要远程操作计算机的情况。许多技术人员会将部分工作带回家里继续进行，这样有时就会需要在办公室登录到家里的计算机同步一些资料；或者

在做技术支持的时候，由于客户往往分散在各地，就需要远程登录到其系统查看系统中出现问题的根源，以便解决。本节将介绍一些远程登录的方法，如远程桌面、SSH 安全登录等。

11.8.1 Vino 远程桌面

远程桌面，就是将另一台计算机的桌面显示在当前计算机上，可以像操作本地计算机一样操作远程的计算机桌面，如浏览网页，运行程序等。如果设置了足够的权限而且网速也够快，则完全与操作本地计算机没有什么区别。

Vino 是 Ubuntu 系统默认的远程桌面系统。要查看远程计算机的桌面，该远程计算机需先启动远程桌面服务。启动远程桌面服务的步骤如下所示。

（1）依次选择“系统”|“首选项”|“远程桌面”命令，出现图 11.1 所示的对话框。



图 11.1　设置远程桌面

（2）选择“允许其他人查看您的桌面”复选框就开启了远程桌面服务。另一个计算机就可以在终端中输入命令 vinagre mylaptop.westnorth-laptop:0 来查看当前计算机的桌面内容了。



使用主机名进行连接的方式，只适合可以正常解析该主机名的情况。一般情况下直接使用对方的 IP 地址更为可靠些，如 vinagre 192.168.1.111:0。可以使用 ifconfig 命令来查看当前计算机的 IP 地址。

（3）如果选择“允许其他用户控制您的计算机”复选框，就会让登录到远程桌面的其他用户对系统的操作具有完全的权限。如果被恶意使用，可能会造成隐私泄露、系统被破坏的严重后果。所以，通常情况下，应该取消选中该选项，这样远程登录的用户就只能观察当前用户的操作，而不能控制系统。

（4）“要求用户输入此密码”复选框被选中，则在其他用户试图连接到该远程桌面时，会提示要求输入密码，密码输入正确才可登录到桌面。这样也大大提高了安全性。如果选择了该项，就需要在“密码”文本框中输入远程桌面的密码。



如果要进行其他设置，可以选择“高级”标签，打开远程桌面高级选项，如图 11.2 所示，进行相关设置。

下面介绍登录远程桌面的方法。选择主菜单“Application”（应用程序）|“Internet”（互联网）|“Remote desktop viewer”（远程桌面查看）命令，即可打开远程桌面察看器，其界面如图

11.3 所示。登录的具体步骤如下。

（1）单击工具栏中 Connect 按钮，弹出对话框。

（2）在弹出的对话框 Host 文本框中输入远程桌面服务器的 IP 地址与端口号，如图 11.3 所示。

（3）单击 Connect 按钮，如果服务器端设置了密码，按提示输入密码即可连接到到远程桌面。



图 11.2　远程桌面高级选项

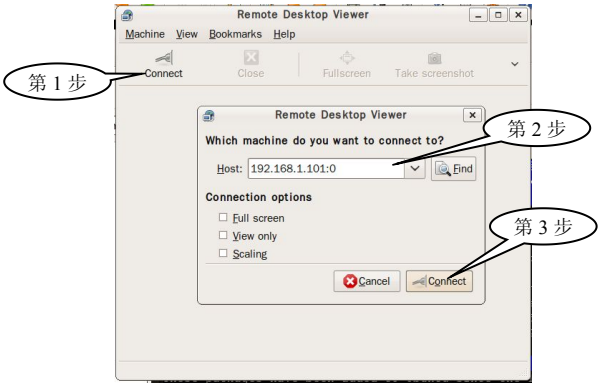


图 11.3　连接到远程桌面

11.8.2　VNC 远程桌面

VNC 也是一种常用的远程桌面连接协议。VNC（Vitual Network Computing）像 X Window 一样，采用了服务端与客户端的模式。下面简单介绍其安装、设置以及使用的方法。

11

1．安装

由于 VNC 采用了客户端与服务器端模式，所以要提供 VNC 服务的计算机应该安装服务器端软件 vnc4server，而要使用 VNC 服务的客户端需要安装 vncviewer，其方法分别如下：

```
sudo apt-get install vnc4server          #服务器端软件的安装方法
sudo apt-get install vncviewer           #客户端软件的安装方法
```

2．设置（服务器端）

为了安全起见，也要给 VNC 服务器设置访问密码。设置密码的命令是 vncpasswd，其方法如下：

```
vncpasswd                #在终端中输入 vncpasswd，以设置密码
   password              #出现该提示时，输入密码
   verify                #重复输入密码，以防出错
```

3．启动 VNC server（服务器端）

设置密码后，就可使用命令 vnc4server 启动 VNC 服务，命令输入后显示如下：

```
earthfans@mylaptop:~$ vnc4server
You will require a password to access your desktops.
#第一次运行该命令会提示输入 VNC 的密码，在 password 与 verify 后输入相同的密码即可
Password:
Verify:
#下面的输出信息很重要，应该记下来。尤其是主机名后的号码(本例中 mylaptop 后的号码是 1)，这是连接到 VNC 服务的重要标志
New 'mylaptop:1 (earthfans)' desktop is mylaptop:1
```

```
Creating default startup script /home/earthfans/.vnc/xstartup
Starting applications specified in /home/earthfans/.vnc/xstartup
Log file is /home/earthfans/.vnc/mylaptop:1.log
```

4．使用 vncviewer 登录远程桌面（客户端）

vncserver 启动成功后，就可以使用 vncviewer 来远程控制桌面了，其用法如下：

```
vncviewer xxx.xxx.xxx.xxx:display number
```

该用法的示例如下：

```
vncviewer 192.168.1.123:1          #IP 地址后面的这个 1，就是在启动 VNC 时得到的号码
```

然后按要求输入密码就可以看到远程的桌面了。不过默认情况下，登录成功的远程桌面只是显示了一个终端窗口，而且相对来说界面很简陋。这是因为 VNC 默认是使用 twm 来作为桌面管理器的。

5．更改 VNC 服务器的桌面管理器（服务器端）

如果系统中安装有其他桌面管理器，可以使用安装的桌面管理器来替代默认的 twm。更改方法是打开服务器的“～/.vnc/xstartup”文件，将最后一行“twm &”修改为“fvwm -r”（如果系统中安装了 FVWM），然后重新启动 VNC 服务。现在重新登录远程桌面后，会发现远程桌面变得美观起来，如图 11.4 所示。

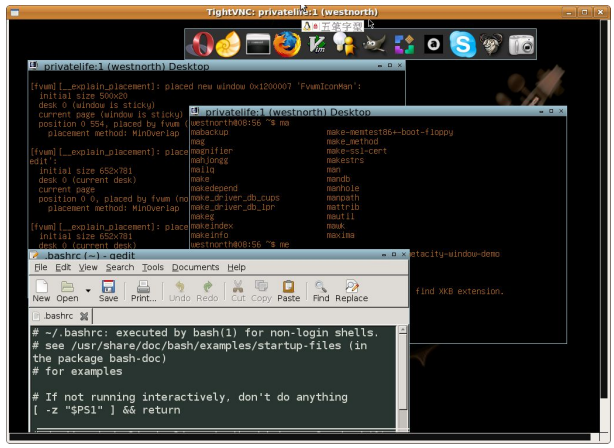


图 11.4　远程登录到使用 FVWM 桌面管理的 VNC 服务器



Vino 与 VNC 均是比较成熟的远程桌面系统。不过，与 Vino 远程桌面比较，VNC 登录方式更为灵活。如 VNC 服务器运行的是 GNOME 桌面，为了节省网络传输开销，可以将 VNC 服务器的桌面管理器设置为轻量级桌面系统。而系统自带的 Vino 只能登录到用户当前所用的桌面系统，这有时会带来某些性能问题。

11.8.3　Telnet 登录

早期计算机在互联时，通常是使用 Telnet 命令来实现远程登录。Telnet 允许远程计算机作为一个用户登录进本地计算机。登录成功后，对于客户机而言，在 Telnet 中进行操作与在普通的终端中操作没有任何区别。如果要计算机提供 Telnet 服务，可以运行以下命令安装两个守护进程：

```
sudo apt-get install xinetd telnetd
```

然后，网络中的其他计算机就可以通过 Telnet 命令来连接到该计算机了。下面是远程登录 Telnet 的示例。

（1）使用 Telnet 命令登录到 192.168.1.101，使用如下命令：

```
earthfans@mylaptop:~$ telnet 192.168.1.101
#下面是欢迎信息
Trying 192.168.1.101...
Connected to 192.168.1.101.
Escape character is '^]'.
Ubuntu 8.04
```

（2）使用 Telnet 登录到远程计算机后，会提示要求输入用户名进行登录。此时应该输入远程计算机中可用的用户名，具体操作如下所示：

```
privatelife login: westnorth        #输入用户名
Password:
#输入密码，这里输入的用户名与密码是主机中实际存在的用户与相应的密码
#如果用户名与密码被接受，会显示如下欢迎信息
Last login: Tue Dec  2 10:09:37 CST 2008 from 192.168.1.101 on pts/3
Linux privatelife 2.6.24-16-generic #1 SMP Thu Apr 10 13:23:42 UTC 2008 i686
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have new mail.
```

（3）登录成功，就可执行远程计算机的系统命令了。如输入 ls 会显示远程计算机当前目录中的文件。

```
westnorth@11:19 ~$ ls
#登录成功，显示 shell 提示符，就可以像在普通的终端中一样执行任意操作了
#本例输入 ls，下面的内容是远程计算机当前目录的文件列表
123             googlehostedservice.html  reder.lst
2               history          rederr.lst
a               hs_err_pid8384.log       rtorrent_guide
a1              html          screenshot
a2              include
```

11.8.4　SSH 登录

Telnet 登录虽然简便易行，但是其存在着一定的安全隐患。因为使用 Telnet 连接的主机和客户机在传输过程中，并没有对传输的内容施加任何的保护。这就相当于在一个透明的管道中输送信息，任何想要截获信息的人均可以拦截信息。为了增强远程登录的安全性，提出了 SSH

（Secure Shell）的概念。

SSH 在传输时，创建了一个加密通道，这样在通道中传输的内容就会被加密。即使有人截获了信息，但由于无法解密也无法得知传输的内容。下面将介绍一些 SSH 服务器架设、登录及文件传输的相关知识。

1. SSH 服务器的架设

对于 Ubuntu 来说，架设 SSH 服务器是一个很简单的任务。一个简单的 SSH 服务器的安装与配置过程如下。

（1）安装 SSH 服务器软件。这是首先要做的工作，这里选用了 openssh-server 软件作为 SSH 服务器。安装命令如下：

```
sudo apt-get install openssh-server
```

（2）安装成功后，在服务器端运行如下命令启动 SSH 服务：

```
sudo /etc/init.d/ssh start
```

（3）在 SSH 服务实际使用前，还应该对 SSH 进行适当的设置，以达到最佳的性能。其配置文件是“/etc/ssh/sshd_config”，部分内容如下所示：

```
cat /etc/ssh/sshd_config     #使用 cat 命令查看配置文件内容
# What ports, IPs and protocols we listen for
Port 22              #采用的端口号，为了安全，可以对其进行更改
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2               #采用 SSH 协议的版本
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes
# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 768
…
```

可以参照上面的说明修改配置文件来更改 SSH 服务器的特性。要注意的是，更改配置文件后，应该重启服务才可使更改生效。重启服务的命令如下：

```
sudo /etc/init.d/ssh resart
```

2. 登录到 SSH 服务器

在一个 SSH 服务器上拥有可用的用户名后，就可以进行 SSH 登录了。下面是一个实际的登录过程。

（1）使用 SSH 命令开始登录到 SSH 服务器 192.168.1.101，假设该 SSH 服务器有一个名为 westnorth 的用户。SSH 命令是系统默认安装的。

```
westnorth@11:35 ～/linux$ ssh westnorth 192.168.1.101
```

```
#使用用户名 westnorth 登录到 192.168.1.101 服务器中。下面是提示信息
The authenticity of host 'westnorth (127.0.0.2)' can't be established.
RSA key fingerprint is 1d:ea:95:a9:bd:d6:94:46:aa:f9:40:5f:1e:79:85:24.
```

（2）系统会询问是否继续连接，回答 yes 到下一步。

```
Are you sure you want to continue connecting (yes/no)? yes #回答 yes
Warning: Permanently added 'westnorth' (RSA) to the list of known hosts.
#将用户 westnorth 添加到信任列表中
```

（3）现在 SSH 服务器会要求输入 westnorth 的密码，输入正确的密码即可完成登录。

```
password:                    #输入密码
```

（4）输入正确的密码后，出现 Shell 提示符，可以像在本机终端上一样进行各种操作。

```
westnorth@11:42 ～/linux$ ls     #出现 Shell 提示符，可以进行相应的操作
…
```

3．分发 SSH 密钥

虽然 SSH 是一个安全的加密通道，但如果不进行设置就无法防止服务器端的欺骗。如果一台计算机使用了伪造的 IP 地址来伪装自己是服务器，那么尽管通信过程中进行了加密，但却将所有的信息发送到了错误的目标。为了防止这种情况的发生，应该在服务器上分发密钥来验证服务器。分发 SSH 密钥的步骤如下。

（1）在分发密钥前，先应该生成密钥，生成密钥的命令是 ssh-keygen，在终端中输入如下命令开始生成密钥：

```
earthfans@mylaptop:～/.w3m$ ssh-keygen -d
#ssh-keygen 是用来建立 SSH 连接所需密钥对的命令，加-d 参数则生成 SSH2.0 版的密钥
Generating public/private dsa key pair.
#生成公/私钥对
```

（2）输入要保存密钥的目录，默认是用户主目录，也可以自行指定位置。

```
Enter file in which to save the key (/home/earthfans/.ssh/id_dsa): /home/
earthfans/.ssh/blog_dsa
#指定密钥对存放的地点，本例选择的目录是/home/earthfans/.ssh/blog_dsa
```

（3）输入 SSH 密钥的密码，这是用来给密钥加密的，要解密也要同样的密码，所以应将密码牢记。

```
Enter passphrase (empty for no passphrase): password
#输入密码
Enter same passphrase again: password
#重复输入同一个密码
```

（4）SSH 显示密钥的保存目录，以及密钥指纹信息、随机图像等信息，如下所示：

```
Your identification has been saved in /home/earthfans/.ssh/blog_dsa.
#用户的私钥保存在 blog_dsa
Your public key has been saved in /home/earthfans/.ssh/blog_dsa.pub.
#公钥保存在 blog_dsa.pub
```

```
The key fingerprint is:
#SSH 的公钥指纹如下
66:be:79:3e:2x:ce:be:20:f2:f5:14:60:42:5q:af:ae earthfans@mylaptop
The key's randomart image is:
#SSH 公钥随机艺术图像如下
+--[ DSA 1024]----+
|    o..          |
|   . . .         |
|    . o .        |
|     0 o         |
|      . 8        |
|     . + .       |
|  . . + o  .     |
|   o + =.o+      |
|    F  oO=..     |
+-----------------+
```

（5）在 SSH 服务器用户主目录中，建立一个名为.ssh 的目录，命令如下：

```
mkdir ～/.ssh
```

（6）将公钥 blog_dsa.pub 放入上一步建立的目录。

（7）使用 chmod 命令修改文件的访问权限为仅 root 可读、写，命令如下：

```
chmod 600 blog_dsa.pub
```

这是因为该文件是属于高度机密文件，应该仅使其被管理员读、写，同时这也是 SSH 服务器的要求。

（8）将密钥文件的名称更改为 authorized_key。

```
mv blog_dsa.pub authorized_key
```

这同样也是 SSH 服务器的要求。

（9）再使用 SSH 命令登录到服务器，会提示说验证文件还未通过验证，是否要继续，输入 yes 继续。然后 SSH 会将验证信息添加到..SSH 目录的 knowhosts 文件。这样，在以后的连接时，就可以一种非常安全的方式连接到 SSH 服务器。两台计算机之间的通信均通过 SSH 加密，这样就不会导致一些敏感信息泄露、造成不必要的损失。

以上步骤（1)～（4）是生成密钥的过程，而步骤（5)～（8）是将密钥分发给服务器的过程。参照如上步骤就可成功地将密钥分发给服务器，实现安全登录。

4．文件的加密传输

建立了 SSH 安全传输通道后，就可以利用该安全通道做许多事，如建立 FTP 连接、加密 HTTP 等。不过，对于普通用户而言，最主要的用途是进行文件传输。由于 SSH 在传输时会对数据进行加密压缩，所以其传输速度还是比较理想的。在 SSH 服务器与本地机之间传送文件的方法如下。

- scp -r westnorth@192.168.1.102:/home/westnorth/scilab5.0.bin ～/software：功能是使用用户名 westnorth 登录到 SSH 主机 192.168.1.102，并将/home/westnorth/scilab5.0.bin 文件下载至本机的主目录下 software 目录。

- scp -r ～/software/maxima.bin westnorth@192.168.1.102:/home/westnorth/：功能是使用用户名 westnorth 登录到 SSH 主机 192.168.1.102，并将“～/software/maxima.bin”文件上传至远程主机的“/home/westnorth/”目录。
- rsync -v -u -a -delete -rsh=ssh -stats westnorth@192.168.1.102:/home/westnorth/scilab5.0.bin ～/software：功能是从远程主机上下载文件。这与前面所述第一条命令功能相似。不过，其只下载比本地目录新的文件。如果本地存在一个名为 scilab5.0.bin 的文件，就不会再重复下载了。这在进行网络同步时是比较有用的。
- rsync -v -u -a -delete -rsh=ssh -stats ～ /software/maxima.bin westnorth@192.168.1.102:/home/westnorth/：功能是将文件上传到远程主机，同样，只会上传新文件，而不会重复上传。

11.9　小结

本章介绍了 Linux 网络管理的相关知识，包括重要的 iptables 的基础知识以及其他网络管理技能。除此之外，还介绍了远程登录的几种手段。本章的重点内容是 iptables，这是 Linux 网络管理的基础，同时也是本章的难点。iptables 功能强大而且使用灵活，所以学习起来比较困难。可以从基础概念着手，循序渐进学习其高级技巧。对于普通用户而言，远程登录也是比较实用的技能，可以参照相关内容在局域网内建立可实现远程登录的服务器，并进行登录，以熟练掌握该内容。

第 12 章　网络相关软件

可以说，网络的出现和普及改变了人类的生活方式和工作方式。网络使得获取、传播信息变得更容易，使得大规模的远程协作变为可能。正如弗里德曼所说，世界变平了。网络使得物理距离变得不再重要，每个人似乎都获得任何想要的信息，人类似乎已经有能力建成巴别塔。而所有这些工作的实现，无一不依赖于这些网络基础工具：网页浏览、邮件收发、即时通信等。本章将详细介绍这些工具的使用。

12.1　Firefox 网络浏览器

Firefox 无疑是 Linux 世界网络浏览器市场的巨擘，其是自由的开源软件，可在多个平台使用，而且功能强大、可扩展性强，已经占领了越来越多的市场份额。本节将介绍 Firefox 的一些基本特色，以及进阶的设置技巧。

12.1.1　Firefox 特色

2002 年，戴夫·海厄特（Dave Hyatt）与布雷克·罗斯（Blake Ross）开发了一个 Mozzilla 的分支项目，目标是让浏览器轻盈、灵活、可扩充。2004 年 2 月 9 日，该浏览器被正式命名为 Mozilla Firefox。由于它具有很多优点，所以得到了很快的发展，至 2009 年 4 月，其在欧洲地区的市场占有率已经超过 IE7，成为主流浏览器的一种。

Firefox 之所以发展如此迅速，当然不仅因为它作为一个挑战者的面目出现，赢得了许多人感情上的支持，更重要的是其提供了在工作与生活中极为有用的一些功能，所以其发展越来越快。目前，其市场占有率已经达到近 30%，并有不断提高的趋势。其主要优点如下。

- 安全性能比较好，这对于 Windows 的用户来说感觉比较明显。Firefox 可以轻易避免主页被修改、设置被锁定等情况的发生。
- 丰富的插件。利用插件可以完成许多看似不可能的任务，极大地丰富了 Firefox 的功能。插件功能也是 Firefox 最大的优势。
- 广告拦截。Firefox 集成了一个拦截弹出式广告的功能，可以自定义设置拦截的网站等。
- 使用 Pango 渲染方式，大大提高了网页加载速度。
- 优良的跨平台支持特性。Firefox 对 Windows、Linux、Mac 等平台都提供良好的支持。

不可讳言，Firefox 还是存在着一些问题的。作为使用者，可以将意见与建议提交给 Mozilla，帮助其改进 Firefox 的功能。Firefox 的一些显著缺点如下所示。

- 启动速度比较慢：当然这是由于所安装插件的原因。
- 运行时内存占用大：内存占用才是 Firefox 的软肋，1.0 版就存在，到 3.0 版仍没有很好

地解决这个问题。

- 每次升级后都会有许多插件无法再使用：这说明其插件接口更换比较频繁，对于用户和开发者而言是一件比较头痛的事。

12.1.2　Firefox 界面

想熟练运用 Firefox，要先熟悉其界面，图 12.1 就是一个典型的运行界面。和其他许多图形界面程序一样，Firefox 也有菜单、工具栏等，下面简要介绍其功能。

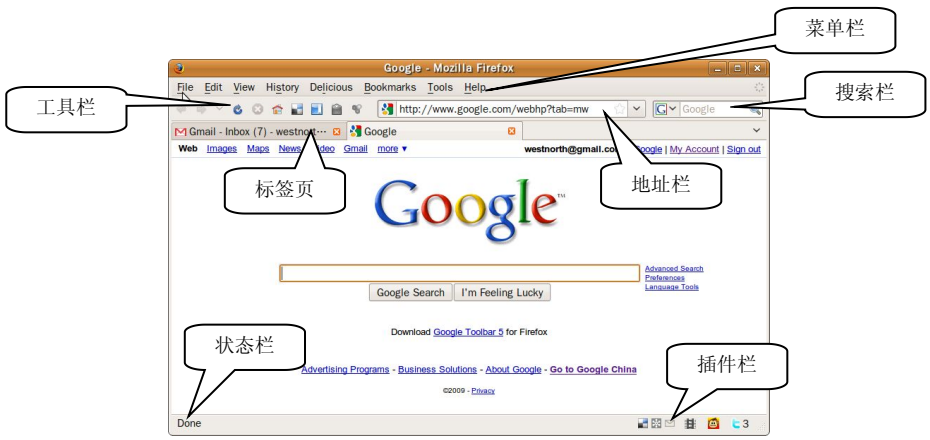


图 12.1　Firefox 界面

- 菜单提供了基础的设置功能，如字体设置、安全设置、书签设置等。
- 工具栏提供了常用的功能按钮，如前进、后退、刷新、回到主页等。
- 地址栏显示当前浏览器打开的网页的地址，也可以在地址栏中输入相关的网址，然后按下 Enter 键打开其他网站。
- 搜索栏提供了一个不进入搜索引擎页面直接搜索的功能。在文本框中输入要搜索的内容，直接按下 Enter 键就可以显示搜索结果了。Firefox 默认的搜索引擎是 Google，但也可以单击小三角符号，选择其他搜索引擎。
- 标签页功能可以使得一次打开多个网页，由于一般用户在网络浏览时均会同时打开多个网页，这也是个很实用的功能。这个功能现在可能算不上什么超炫的功能，但该功能在 Firefox 2.0 时就有，与其同时期的 IE6 就没有标签页浏览功能。
- 状态栏显示打开网页的进度或当前浏览器正在进行的动作。
- 插件栏显示安装的一些插件。这些插件一般是经常使用或需要查看其状态的，如 Gladder，需要经常切换其代理状态，所以在这里显示。

12.1.3　初步设置

Firefox 虽然优秀，但在使用前如果不进行一些基本的设置，可能性能和外观还是无法达到一种让人满意的程序。使用前的基本设置工作一般包括字体的设置、主页的设置，以及使用 about:config 设置等内容。

1．字体设置

默认情况下 Firefox 的字体并不太美观，要让其显示自己想要的字体，就要进行相关设置。设置的步骤如下所示。

（1）在 Firefox 中依次选择 Edit（编辑）|Preference（首选项）命令，打开如图 12.2 所示的对话框。

（2）单击 Content（内容）标签，打开内容页面。

（3）单击 Fonts & Colors 区的 Advanced（高级）按钮，打开 Fonts（字体）对话框，如图 12.3 所示。这里就是设置字体的地方。

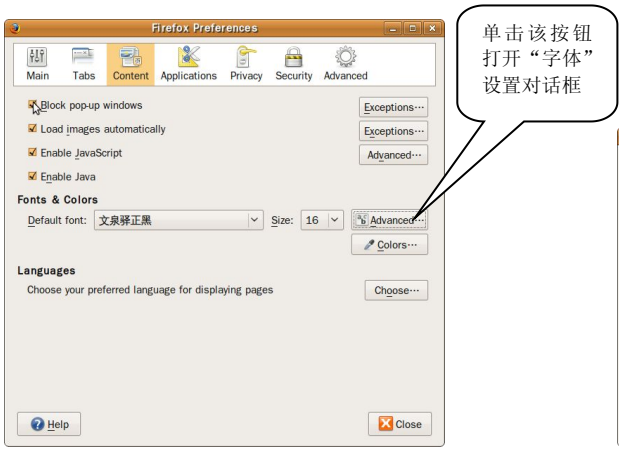


图 12.2　首选项对话框



图 12.3　Firefox 字体设置

（4）将 Fonts for（字符集设定）设置为 Simplified Chinese（简体中文）。

（5）将 Proportional 设置为 Sans Serif（无衬线字体）。

（6）将 Serif（衬线字体）设置为 Bitstream Vera Serif。

（7）将 Sans Serif（无衬线字体）设置为文泉驿正黑。

（8）将 Monospace（等宽字体）设置为 Bitstream Vera Sans Mono。

（9）将 Default Character Encoding（默认字符编码）设置为 Chinese Simplified（GB2312），这样显示中文就会较美观了。

上述步骤（4）～（9）当然也可以根据自己的需要设置。需要注意的是要将 Serif 设置为一款衬线字体，而将 Sans-serif 设置为一款中文字体。关于什么是衬线字体，请查阅第 3.3.4 节相关内容。

2．主页设置

Firefox 的主页设置和 IE 的基本一样。用前述方法打开“首选项”对话框，选择 Main 选项卡，在其中的 Home Page（主页）文本框中填入想要的主页地址，如 http://www.google.com，下次打开时即可默认打开该网页。

Firefox 的主要选项均在图 12.2 所示对话框中进行设置，在需要时可以依次打开各标签页进行查看，而且设置均较为直观，这里就不再列举了。

3．配置文件设置

在浏览器地址栏中输入 about:config，就打开了 Firefox 的配置文件设置功能。在其中可以

对 Firefox 的运行方式进行设置。在打开配置文件之前，Firefox 会显示一个提示页面，意思是这项操作会有风险，操作需谨慎之类，单击“I’ll be careful ，I promise”（我保证我会小心）按钮，就可以进入了。打开后，里面列出了一行行的内容，一行代表一个参数，可以单击其进行设置。但这些值一般会让人感到不知所措，所以将部分常用值列举如下。

- accessibility.typeaheadfind.Enablesound：True 意思是没找到要找的内容时发出声音，False 是不发声。
- accessibility.typeaheadfind.Linksonly：True 是只搜索超级链接上的文字，False 是搜索所有页面内容中的文字。
- advanced.always_load_images：总是加载图片。
- applications.tmp_dir：临时文件夹的目录。
- browser.active_color：超级链接的颜色。
- browser.chrome.favicons：地址栏显示网站图标。
- browser.chrome.site_icons：收藏夹显示网站图标。
- browser.display.screen_resolution：Firefox 的屏幕分辨率。0 代表使用系统的分辨率。
- browser.fixup.alternate.enabled：Firefox 是否自动修正地址的输入。如将 http://google / 修复成 http://www.google.com，其中前缀 www 和后缀 com 是可以自由定义的。
- browser.fixup.alternate.prefix 和 browser.fixup.alternate.suffix 定义 Firefox 修复地址时的前缀和后缀。两项默认值是 www.和.com，可以修改为自己想要的值。
- browser.formfill.enable：保存历史记录。
- mousewheel.withnokey.numlines ：鼠标滚动时页面移动的行数。
- plugin.default_plugin_disabled：是否提示安装插件。

除了 about:config 外，Firefox 还有许多其他的 about，如下所示。

- about:buildconfig：显示编译时所用的选项。
- about:cache：显示缓存目录。
- about:mozilla：Mozilla 之书的某章节。
- about:plugins：显示已安装的插件信息。

4．实现加密代理

有时，上网浏览时 Firefox 会出现“无法连接到服务器”或“连接被重置”的错误提示，无法访问某些网址。这并不是浏览器的问题，而是由于种种原因，用户无法访问某些网址。这时，可以采取一种叫做代理服务器的技术来访问该网址。这样，一般就可以打开该网页了。

代理可以使用下文介绍的 Gladder 插件，但该插件有时不太稳定，而且在读取和发送信息时有可能存在安全问题。Ubuntu 还提供了 Stunnel 软件来实现加密代理。

Stunnel 可以用来做 HTTP、Socket 等代理。安装很简单，由于在 Ubuntu 软件库中有此软件，所以使用命令 sudo apt-get install stunnel 即可进行安装。



代理的原理很简单。假设用户是 A，要访问 C 网站，但由于种种原因无法连接。又有一个用户 B，该用户可以正常访问 C 网站。这时，就可以将 A 通过一些技术手段连接到 B 上，通过 B 的中转来让 A 访问网站 C。这时，称 B 为代理服务器。

Stunnel 安装成功后，就可使用其来实现加密代理了。具体步骤如下所示。

（1）使用如下命令打开 Stunnel 客户端：

```
sudo stunnel -c -d 127.0.0.1:7878 -r 193.35.228.209:443
```

意思是使用本机 7878 号端口作为代理服务器连接的端口，端口是用-d 参数指定的。而使用的代理服务器地址是 193.35.228.209，代理服务器端口是 443。参数-c 的意思是使用客户端功能。同时 Stunnel 还可用做代理服务器。

（2）现在开始在 Firefox 中设置代理。打开 Firefox 菜单，选择 Edit|Preference 命令，出现如图 12.4 所示的对话框。

（3）单击 Advanced 按钮，出现高级设置页面。

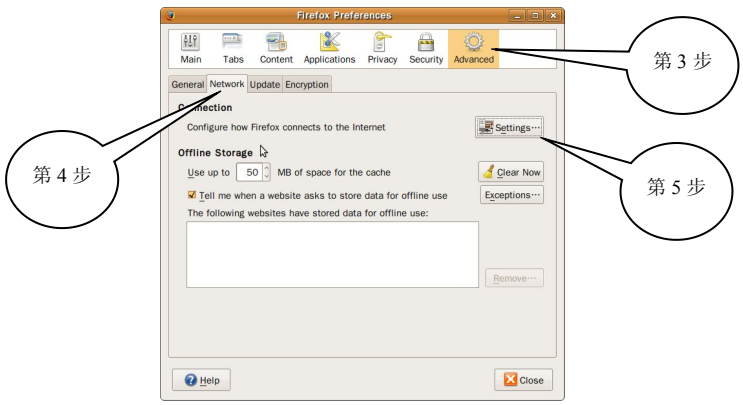


图 12.4　Preference（首选项）对话框

（4）选择 Network 标签，打开网络设置选项卡。

（5）单击 Settings 按钮，出现如图 12.5 所示对话框。

（6）选择 Manual proxy configuration（手动设置代理）单选按钮。

（7）在 HTTP Proxy（HTTP 代理）文本框中输入 127.0.0.1。

（8）在 Port 微调按钮中输入 7878。



第 7、8 项的设置值均为第 1 项命令使用的值。如果命令中的值发生了改变，在设置 7、8 项时也应该做相应的改变。

（9）选择 Use this proxy server for all protocols（所有协议均使用该代理）复选框，让所有的联网服务全部使用该代理。

（10）单击 Finish 按钮结束设置。

（11）随便打开一个查询本机 IP 地址的网站，可以看到，本机的 IP 地址已经成为 193.36.228.209，而不是电信分配的哪个 IP 地址了。

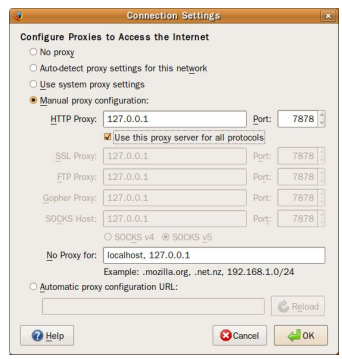


图 12.5　代理设置

12.1.4　安装插件

Firefox 的强大主要体现在其可扩展性，实现可扩展性主要通过安装各种插件。本节内容主要介绍几个常用且

功能强大的插件。

1．加快下载速度的插件 DownThemAll

Firefox 有下载管理软件，但速度其实还真是不敢恭维。所以就有人开发了一个强大的下载管理插件 DownThemAll。安装其实很简单，保证处于联网状态，然后参照如下步骤进行即可。

（1）选择 Firefox 菜单中的 Tools|Add-ons 命令，弹出的对话框如图 12.6 所示。

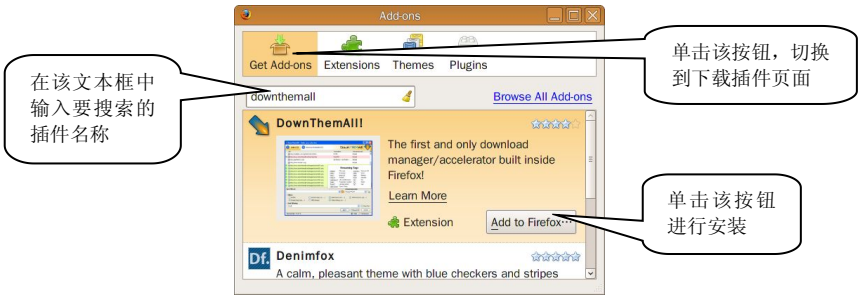


图 12.6　搜索插件对话框

（2）单击顶端的 Get Add-ons 按钮，切换到下载插件页面。

（3）在文本框中输入要下载的插件名称，本例输入 downthemall。输入后按下 Enter 键，Firefox 就会在其庞大的插件库中搜索出输入项匹配的插件，并将结果列在列表栏中。

（4）图 12.5 列出了 DownThemAll 的插件，单击该项的 Add to Firefox 按钮开始安装过程。

（5）在开始安装安装前，Firefox 会提示用户安装插件时的注意事项，不用理会，直接在图 12.7 所示对话框中单击 Install 按钮（要等待其后的数字减小为 0）即可。

（6）插件安装完成后提示要重启 Firefox 以使插件生效，单击 Restart（重启）按钮重启后此插件生效。

（7）使用也很简单。当用户单击了特定的文件（如 zip 压缩文件、MP3 音乐文件、mpeg 视频文件等）时，系统会弹出一个如图 12.8 所示的对话框，询问用户如何处理该文件。



图 12.7　安装插件



图 12.8　使用 DownThemAll 下载

（8）选择第 1 项（DownThemAll），然后单击 Save File 按钮，打开如图 12.9 所示的对话框。

（9）在 Save files in 文本框中设置保存文件的位置。

（10）如果有 MD5 码，在 Checksum hash 文本框中输入 MD5 码，下载完成后会自动进行校验。

（11）单击 Start！按钮，DownThemAll 就会开始高速下载。

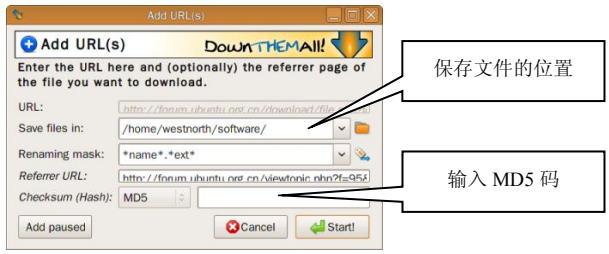


图 12.9　下载设置对话框

2．支付宝插件

由于原来支付宝网站的安全控件使用了 Microsoft 的 Active 技术，所以直接导致 Linux 平台的用户无法使用支付宝。2008 年 10 月，支付宝终于发布了针对 Linux 平台的支付宝安全插件，这个插件暂时没有提交到 add-on 列表，所以无法通过第 1 种方法进行安装，要手工进行，比较麻烦一点，不过还是物有所值的。

安装前，先在 http://blog.alipay.com/wp-content/2008/10/aliedit.tar.gz 下载插件的压缩包。由于其是很重要的安全插件，所在建议在安装前校验其 MD5 码值，这个版本的 MD5 是 02240053d32688ed996e3a4788042801。安装步骤如下（以下步骤均在终端中进行）。

（1）进入该包所在的目录，如下载的插件压缩包位于～/software，可用命令“cd download”。

（2）使用命令“md5sum aliedit.tar.gz”检验其 MD5 码是否等于前文给出的值。

（3）用“tar -zxvf aliedit.tar.gz”解压插件压缩包，解压出来共有两个文件，aliedit.so、aliedit.xpt。

（4）用“cp aliedit* ～/.mozilla/plugins”将支付宝插件的这两个文件复制到 Firefox 插件目录中。



如果～/.mozilla/plugins 目录不存在，则使用命令“mkdir ～/.mozilla/plugins”建立。

（5）重新启动 Firefox 即可正常使用支付宝的各项功能，如查询、付款等。

3．其他常用的插件

下面所列的插件均包含在 add-on 列表中，可以通过前述安装 DownThemAll 的方法进行安装。

- Adblock Plus ：拦截弹出窗口。
- ColorfulTabs ：让每个标签页显示不同的颜色。
- del.icio.us Bookmarks：delicious 网络书签，功能极其强大，可以方便地将网址加入其中，也可以快速寻找到书签中的网址。
- Drag de Go ：在 Firefox 中使用鼠标手势。
- Gmail Space：将 Gmail 作为网络硬盘使用。
- Google Toolbar：比较强大，有书签功能，有网页“即指即译”功能。
- Google notebook：方便地使用 Google 笔记本。
- Greasemonkey ：使用自定义的脚本来重定义 Firefox 的默认行为与显示格式（推荐）。
- Add to Search Bar：可以自定义使用哪一个搜索引擎进行搜索，如 verycd、douban 等。
- CHM Reader：Linux 不能直接读取 chm 文件，使用此插件就可以将 chm 文件在 Firefox 中打开。

- VideoDownloader：可以下载网页中正在播放的 Flash 视频，如 Youtube 等大视频网站中的视频。
- Gladder：网络代理软件。安装后在插件栏会出现一个梯子的图标，右击该图标，在弹出的快捷菜单中选择 Proxy current page（使用代理打开当前页面）命令，就可自动使用代理打开当前页面。或者选择 Option 命令，将当前网址加入 Personal Banned Sites List 可实现打开该网址时自动代理。如按图 12.10 所示的设置，则可在打开 *.wikibooks.org 等网站时自动使用代理。



图 12.10　Gladder 的选项

12.2　精致小巧的 Opera 浏览器

Firefox 在 Linux 中俨然有王者之势，但由于其存在的问题，所以给了 Opera 一定的发展空间。Opera2.1 版是首次公开发行，时间是 1996 年。Opera 在个人电脑市场占有率一直是微乎其微的，但在手机和各种移动设备中市场占有率极高。现在，由于其拥有的一系列优点，在个人电脑中也开始了小规模的流行。

由于软件源中没有提供 Opera，所以要到如下地址进行下载，下载地址是 http://www.opera.com/download/linux/。其主要特点是：

- 速度快：Opera 利用了缓存机制，使得载入页面速度比其他软件有明显的提高；
- 功能丰富：Firefox 只拥有单纯的网页浏览功能。Opera 则拥有邮件收发、IRC 聊天、RSS 阅读功能等；
- 灵活性高：Opera 使用了快速拨号来方便使用，还支持鼠标手势功能；
- 其他优点如完美的可定制性、高安全性等。

1．软件界面

打开 Opera 后，默认界面如图 12.11 所示。其界面其他元素与 Firefox 等浏览器基本相同，就不再赘述。下面简单介绍一下 Opera 的特色功能快速拨号。

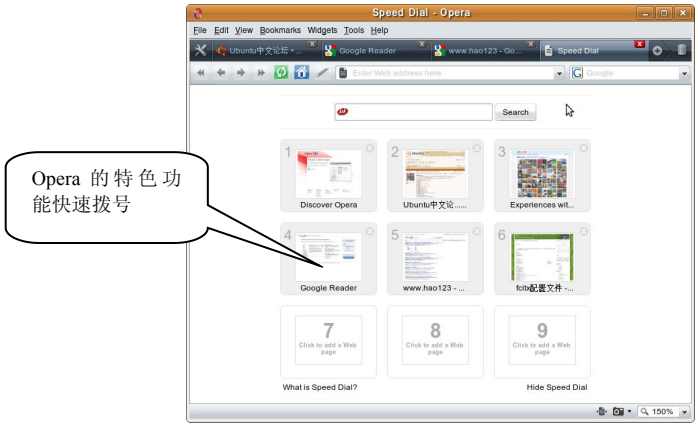


图 12.11　Opera 主界面



图 12.12　添加快速拨号的网址

快速拨号从图 12.11 可以看出其基本界面。所谓快速拨号就是在 Opera 的主界面显示了 9 个方格，每个方格显示了一个网页的缩略图。我们将每一个方格称为一个键。单击显示了缩略图的键就可打开相应的网页。如果相应的键是空白的，如图 12.11 所示的 7、8、9，说明该键还没有相对应的网址。单击该键，会弹出相应的添加网址的对话框。本例按下 7 键，弹出对话框如图 12.12 所示。

现在可以在列表中单击任一网址将其加入该键，也可以在图示文本框中输入网址。这样，设置完成后，按下 7 键就可打开相应的网页。

2．字体设置

Linux 版 Opera 使用中的第一个重要问题就是中文字体显示。默认情况下，Opera 的中文显示效果很差，有时甚至会出现字体大小不一的杂乱版面。所以要让 Opera 正常使用，首先就要设置字体，使其正常显示。设置字体的步骤如下所示。

（1）依次选择 Opera 的菜单 Tools|Preference 命令，打开 Preference 对话框，如图 12.13 所示。

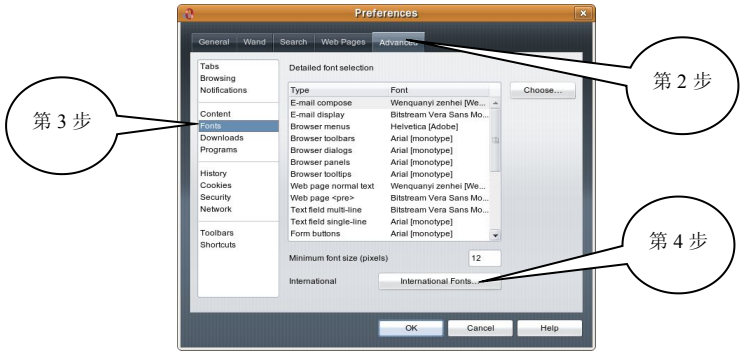


图 12.13　Opera 字体设置

（2）选择 Advanced 标签，打开高级设置页面。

（3）选择页面左侧列表栏中 Fonts 选项，打开字体设置页。

（4）单击 International Fonts 按钮，打开国际字体设置对话框，如图 12.14 所示。将其中的 Writing system 下拉列表框更改为 Chinese simplified（简体中文），然后将字体更换。



图 12.14　国际字体设置对话框

但很多用户都反映，这样设置后，中文字体显示仍然不正常。对于这个问题，官方的解释是字体替换问题，给出的解决方案是这样，输入命令“sudo gedit /usr/share/opera/ini/font.ini”，将其中内容全部删除，然后输入如下内容：

```
Opera Preferences version 2.1
```

```
; Do not edit this file while Opera is running
; This file is stored in UTF-8 encoding
[matches]
; One can disable xft or core fonts if necessary
;engine:xft=blacklist
engine:x11=blacklist
; Foundries that are generally known to have bad implementations
;foundry:urw|abiword|ultimo=bad
; Generic settings - families with known glyph types
family:times|nimbus roman no9 l|times new roman|bitstream vera serif=serif
family:helvetica|nimbus sans l|arial|bitstream vera sans=sans-serif
family:courier|courier new|bitstream vera sans mono=monospace
; Generic settings - generic families which have good, readable designs,
; but where we can't be sure the implementation is any good.
family:times|times new roman=good
family:helvetica|arial=good
family:courier|courier new=good
family:clearlyu *=good
; Known high-quality fonts
family:arial|verdana|times new roman;foundry:microsoft|monotype=excellent
family:Nimbus*=verygood
foundry:bitstream=excellent
; Known fonts that solves specific problems
;family:mincho|gothic=japanese good try-first
;family:kochi*=japanese good try-first
family:AR PL UMing*=chinese-s good try-first
family:wenquanyi zenhei=chinese-s verygood try-first
family:wenquanyi zenhei=chinese-s good try-first
;family:baekmuk*=korean good try-first
; Fonts that we know should not be used for displaying text
family:*dingbat*|*dings*|agathadaimon=symbol
; Specific fonts at the bottom
; Known bad implementations
engine:x11;foundry:mdk;family:helvetica=blacklist
[default]
usage=by-request
preference=5
[serif]
style=serif
[sans-serif]
style=sans-serif
[monospace]
style=monospace
[cursive]
style=cursive
[fantasy]
```

```
style=fantasy
[japanese]
style=japanese
[chinese-t]
style=chinese-t
[chinese-s]
style=chinese-s
[korean]
style=korean
[try-first]
usage=try-first
[excellent]
preference=8
usage=try-first
[verygood]
preference=7
usage=try-first
[good]
preference=6
[bad]
preference=4
[avoid]
usage=by-request
[symbol]
usage=by-request
[blacklist]
usage=blacklist
```

一般来说，这样就可以解决中文显示的问题，而且没有任何版权问题，因为使用的均是 Ubuntu 自带的自由字体。

3．在线影音播放功能

Opera 使用中还有一个问题就是无法播放在线影音文件。解决方法是将/usr/lib/firefox/plugins/和/usr/lib/mozilla/plugins/目录中的所有文件复制到/usr/lib/opera/plugins/中。其实也就是将 Firefox 的在线影音播放插件复制到 Opera 的插件目录。这样 Opera 就能正确识别音频、视频文件了。

12.3 安装 IE 浏览器

不得不说，在实际的网络浏览过程中，有时还是离不开 IE 的。由于 Windows 的垄断地位，国内许多网站在设计网页时，并没有针对非 IE 核心进行详细测试，导致非 IE 浏览器浏览网页时出现种种问题，如排版错误、无法收听在线音乐（如搜狐的音乐盒），或者干脆就无法使用（如大多数的网上银行服务）。

在这种情况下，Linux 用户有两个选择。一是使用虚拟机安装一个 Windows，另外一种比

较好的选择是使用 Wine，直接在 Linux 中安装一个 IE。由于种种原因，直接安装 IE 是不会成功的。不过，有需求就有市场，一个巴西公司开发出了一个叫 ies4Linux 的程序，可以使用这个程序来将 IE 安装到 Linux 中。

安装 ies4Linux 须使用命令行进行。其实，也不要看到命令行就产生畏惧心理，命令行方式有许多图形界面无法比拟的优点，如使用快捷、功能强大等。安装 ies4Linux 只需要以下四步即可。

（1）选择主菜单应用程序|附件|终端打开终端，然后在其中输入以下命令：

```
wget http://www.tatanka.com.br/ies4linux/downloads/ies4linux-latest.tar.gz
```

这是使用 wget 下载 ies4Linux 的软件包。注意这儿在 ies4linux 后面并不是版本号，而是 latest。使用该文件名下载时，会得到最新的 ies4linux 软件。

（2）解压该软件包，因为下载得到的是一个压缩文件，需要先解压才可使用。解压命令如下：

```
tar zxvf ies4Linux-latest.tar.gz
```

（3）进入软件的目录，解压后得到的目录名应该类似于 ies4Linux-1.203.2。这时，后面的数字是实际的版本号，根据开发状况，这里的版本号会有所不同。输入如下命令：

```
cd ies4Linux-*
```

后面的“*”用实际的版本号代替，一个很简单的方法是输入“cd ies4Linux-”后按下 Tab 键让终端自动补全。一般而言，自动补充的目录就是软件的目录。

（4）软件目录中提供了一个名为 ies4Linux 的可执行文件（实际是一个脚本文件），运行该文件会开始安装。运行的命令如下：

```
/ies4Linux
```

在安装过程中可能需要在微软的网站上下载一些 dll 库文件，所以安装还是需要一定的时间。安装完成后，就会在桌面上出现一个熟悉的 IE 标志，双击其就可使用。其使用方法与在 Windows 中使用并无二致。可以像在 Windows 中一样，安装网上银行的安全控件、浏览 Firefox 无法正常显示的网页。

12.4　字符界面网页浏览软件

许多时候，图形界面软件并不能满足要求。如远程登录到服务器的终端，这样就无法使用图形界面程序；或者如果本机的 X Window 出现问题，用户只能登录到终端中，而要找出问题的解决办法，还得上网查找相关资料，这时，文本方式浏览器就派上了用场。本节介绍了两种常见的文本方式网络浏览器：w3m 和 lynx。

12.4.1　强大的字符界面浏览器 w3m

w3m 是一款强大的文本界面网络浏览器。其使用很简单，在终端中输入 w3m www.google.com 即可开始网络冲浪。由于这是一种字符方式浏览，与普通的图形界面软件有着很大的差别。先来看一张 w3m 浏览网页的截图，如图 12.15 所示。

进入 w3m 后，默认是不支持鼠标操作的，只能使用键盘的快捷键进行相关的操作。下面介绍一些对于浏览网页而言，最为基本的键盘操作技巧。

- 打开链接：单击链接打开网页，一般要移动光标的位置，在无法使用鼠标的情况下只有键盘。w3m 中移动光标位置，可以使用方向键。要将光标从一个超级链接移动到另一个超级链接，可以按下 Tab 键进行切换。移动到要打开的链接后，按下 Enter 键在当前页中打开该链接。如果按下 Ctrl+T 快捷键则在新的标签页中打开目标网页。这也是 w3m 比较酷的地方之一：选项卡浏览。

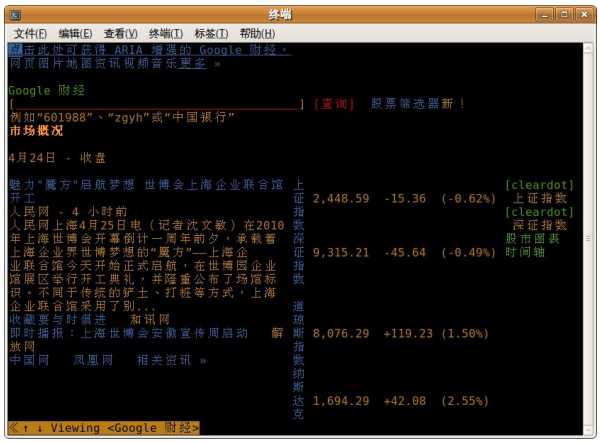


图 12.15　w3m 的界面

- 文本框中输入内容：按下 Tab 键将焦点移到文本框中，然后按下 Enter 键，屏幕下方会出现输入栏，在该输入栏中输入文字，再按下 Enter 键，这些文字会被添加到文本框中。
- 文本搜索：按“/”键在屏幕下文出现输入提示，输入文字按下 Enter 键可进行搜索，还可以按 N 键继续上次的搜索。
- 查看图片：将光标移到图片网址之上，然后按 I（大写字母）键就可以调用外部程序打开图片。
- 页面转移：按下 B 键后退到上一个打开的页面中。按“{”、“}”键可以在选项卡之间转移。
- 历史记录：按下 Ctrl+H 快捷键可以打开访问过网页的历史记录。
- 获得在线帮助：按下 H 键可获得帮助。

前面只是介绍了一些最为常用的快捷键操作，但 w3m 的功能极为丰富，使用快捷键可以完成许多复杂的功能。下面列出比较完整的 w3m 功能键列表。

表 12.1　w3m 功能键列表

功 能 键	功　　能
方向键	移动光标
<、>	向左、向右移动屏幕
w、W	将光标向前、向后移动一个单词
空格键	与 PageDown 一样，向下翻一页，其他如 Home、End 等键也可移动光标
Esc-m	列出所有超链接

Esc-g 行数	跳到某行
Esc-Tab	移动光标到上一个超链接

续表

功 能 键	功　　能
[、]	移动光标到第一个或最后一个超链接
B、Alt-向下方向键	回到上一页
回车	打开超链接
Ctrl-t	在标签页中打开超链接
R	刷新页面
T	开一个新标签页
Ctrl-q	关闭当前标签页
{、}	移动到下一个、上一个标签页
Esc-t	显示所有标签页，以供选择
/、?	向下、向上搜索文本
n、N	继续前次的搜索
o	打开选项设置菜单
H	显示帮助信息
ctrl-h	显示浏览历史记录
q、Q	提示、不提示退出



可用 w3m 来将网页文件转换为纯文本格式。命令是: cat foo.html | w3m -dump -T text/html >foo，意思是先用 cat 显示文件内容，将内容用管道传送至 w3m，用 w3m 转换功能将 HTML 文件转换为文本文件，再将文本的内容重定向到文件 foo。w3m 还可用来作文本显示，如“ls | w3m”、“w3m .bashrc”等。

12.4.2　高速字符界面浏览器 lynx

lynx 是另一个字符界面网页浏览器。lynx 占用资源同样极少，速度同样很快。lynx 的主页地址是 http://lynx.browser.org，可以在 http://www.cc.ukans.edu/lynx_help/lynx_users_guide.html 得到用户指南。lynx 的默认主页由环境变量 WWW_HOME 定义，如果不存在该变量，lynx 则会显示它的主页 http://lynx.browser.org/。

lynx 默认情况下对中文支持不好，而且打开的页面完全没有布局的概念，但个人感觉速度要比 w3m 快一些。lynx 同样由键盘操作，其操作不再详细介绍，只将其罗列于下，以供查阅。

- 链接之间的移动用上、下方向键。
- 退回到上一页，用向左方向键。
- 打开链接地址，用 Enter 键或向右方向键。
-)、(：向下、向上翻半页。
- c：建立新文件。
- d：下载当前链接。
- E：编辑选中的链接。
- r：删除选中的文件。

- H：获得帮助信息。
- a：将当前链接加入书签。
- g：可以打开输入的网址。
- G：编辑当前地址，并打开新的地址。
- k：显示键盘命令列表。
- o：设置选项。
- q：退出。
- /：在当前页内查找字符串。
- n：继续前次的搜索。
- v：查看一个书签文件。
- z：停止加载当前页面。
- =：显示当前页的编码、日期等信息。
- \：显示当前页的源代码。
- !：退出。
- Ctrl+r：刷新页面。
- 数字键：跳转到后面第 *n* 个超链接。

12.5 收发电子邮件

电子邮件在网络使用中是极其重要的一个方面。在很多人的网络使用中，电子邮件是一切的基础。每天工作、生活的重要内容就是在电子邮件程序里管理邮件、管理日程表、管理任务等。Windows 平台中，使用 Office 组件中的 Outlook 可以完成此功能。在 Linux 中，可以使用相应的程序 Evolution 来管理邮件。Linux 还提供了一种字符界面管理邮件的方法，本节也将给予介绍。

12.5.1 图形界面软件 Evolution

Evolution 是 Linux 默认的邮件管理程序。其功能类似于 Microsoft Office 中的 Outlook，可以实现邮件管理、日程管理、任务管理、简单记事等功能。选择“应用程序”|“办公”|“Evolution 邮件和日历”命令，就可打开该程序。

1．建立新的邮件账户

Evolution 程序在使用前需要进行一些简单的设置。设置内容主要包括邮件地址和收发邮件的服务器等信息。设用户拥有的电子邮件地址为 mylazylifeblog@sohu.com。下面介绍具体的设置过程。

（1）首先，Evolution 显示欢迎屏幕，单击“前进”按钮到下一步。

（2）设置标识，在“全名”文本框中输入用户的姓名，在“电子邮件”文本框中输入电子邮件地址，本例使用 mylazylifeblog@sohu.com。单击“前进”按钮到下一步。

（3）服务器类型选择 POP 选项。这是因为 sohu 使用 POP 协议来接收邮件。现在界面如图 12.16 所示。“服务器”文本框中填入 pop3.sohu.com，用户名填写@前的内容。单击“前进”按钮到下一步。

（4）在“接收选项”对话框中，选择“在服务器上保留信件”复选框，这样下载到本地的

信件只是一个备份，而原始邮件仍然保存在服务器中。如果不选这项，Evolution 会在下载邮件的同时将服务器中的原始邮件删除。如果有必要，还可以再勾选“7 天后删除”的选项。单击“前进”按钮到下一步。

（5）设置发信服务器。服务器类型选择 SMTP 选项，在服务器文本框中填入 smtp.sohu.com，注意要选择“服务器需要认证”复选框。设置如图 12.17 所示。单击“前进”按钮到下一步。



图 12.16　设置收件服务器信息



图 12.17　设置发件服务器信息

（6）现在设置本账号在 Evolution 中的名称，可以自己随意填写。单击“前进”按钮到下一步。

（7）现在设置已经全部结束，单击“完成”按钮结束设置。

如果还要再添加其他的邮箱信息，可以单击 Evolution 菜单“编辑”|“首选项”命令，打开如图 12.18 所示的对话框。单击对话框左侧“邮件账户”按钮，然后单击右侧“添加”按钮，就开始了一个新建邮件账户工作。

如果某账户的设置有问题，还可以在图 12.18 中选中该账户后单击“编辑”按钮进行修改。

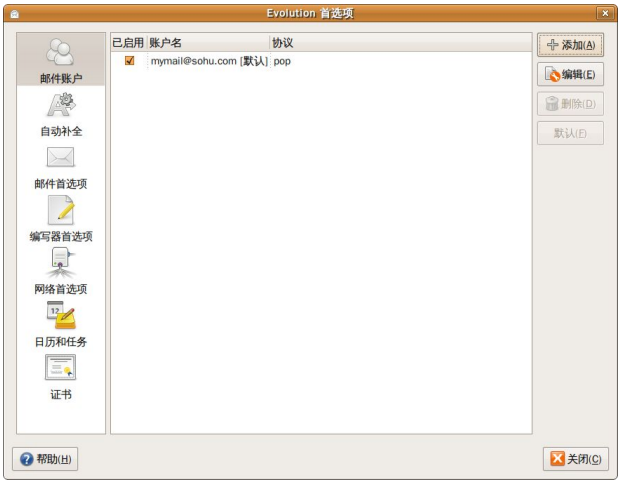


图 12.18　首选项

2．Evolution 的界面

进行完上面的步骤后，就可开始正式的使用过程，如果邮件的地址信息、密码等均设置正确，单击“结束”按钮后，Evolution 应该已经开始接收服务器中的邮件了。Evolution 的界面

如图 12.19 所示，习惯了 Outlook 的读者应该觉得很熟悉吧，界面几乎与 Outlook 一样！

界面除去用户比较熟悉的的菜单和标题栏外，可简单地分为四个区。

- “文件夹”包含了邮件收发中用到的目录，如“收件箱”，“发件箱”，“回收站”等。这些目录可以自己创建。
- “邮件信息列表栏”显示电子邮件的发件人姓名、邮件主题等信息。



图 12.19　Evolution 界面

- “邮件正文”显示邮件的内容。
- “扩展功能”包括其他功能，如日程管理、联系人列表、备忘等功能。

直观的界面让操作也变得轻松起来，单击“新建”工具栏按钮，则是用当前的邮件设置写电子邮件，而“发送/接收”按钮是将本地发件箱中的信件发出去，同时将服务器中邮件收回来。“回复”是给当前正在打开的信件写封回信。“转发”是将当前打开的信件转发给另一人。“垃圾”是将当前打开的信件设置为垃圾邮件，今后该人发过来的信直接会被送到 Evolution 中的“回收站”。

要使用其他功能，则简单地单击“扩展功能”区域的相关按钮，如单击“日历”就可进入日程管理界面，可以在该界面进行日程安排，然后在日程安排时间快到时，会弹出相关提示信息。

12.5.2　字符界面邮件管理

基于与使用字符界面网络浏览器同样的理由，在实际使用中，有许多地方还是需要字符界面的邮件管理器的。Linux 世界中，最为著名的字符界面邮件管理软件是 mutt。据“财富”杂志报道，IT 界的传奇女性、Google 的高级副总裁 Marissa Mayer 使用的邮件管理软件就是 mutt。但是 mutt 的配置过于复杂，所以本节以 Emacs 这一著名文本编辑器中自带的 Gnus 模式来进行邮件管理。

众所周知，Emacs 是一款全功能的强大的文本编辑器，其功能当然包括邮件管理了。其邮件管理是通过 Gnus 来实现的。gnus 其实是一款新闻组阅读器，不过稍做改进，就可用其来管理邮件。其操作简单，功能强大，本节所述只是最为基本实用的使用方法。

1．Gnus 配置文件

要让 Emacs 的 Gnus 开始工作，必须建立 Gnus 的配置文件。下文所述内容可以单独放在

“～/.gnus”文件中，也可以放在“～/.emacs”中，推荐前者。新建“～/.gnus”文件，然后将如下内容加入，注意要在用户名与密码处填入自己的用户名与密码。

```
(require 'smtpmail)
 (require 'gnus-load)
;;设置一个国内的服务器地址，这样可以大大减少 gnus 启动的时间
(setq gnus-select-method '(nntp "news.cn99.com"))
;;开启代理功能，可以使 gnus 离线浏览下载的邮件
 (gnus-agentize)
;; 设置邮件的服务器与用户名密码
 (setq mail-sources
    '((pop :server "pop3.sohu.com"   ;; 在这里设置 POP3 服务器
           :user "mylife"           ;; 用户名
           :port "pop3"
           :password "123456")))  ;; 密码
;设置保存邮件的方式，本次选择 nnfolder
 (setq gnus-secondary-select-methods '((nnfolder "")))
;; 设置 SMTP 服务器
(setq user-full-name "Godfrey Ma")
(setq smtpmail-local-domain "sohu.com")
(setq smtpmail-sendto-domain "sohu.com")
(setq user-mail-address "mylife@sohu.com")
(setq smtpmail-auth-credentials
   '(("SMTP.sohu.com"  ;; SMTP 服务器
     25
     "mylife"        ;; 用户名
     "123456")))     ;; 密码
(setq smtpmail-default-smtp-server "SMTP.sohu.com")
(setq smtpmail-smtp-server "SMTP.sohu.com")
(setq message-send-mail-function 'smtpmail-send-it)
;中文支持
(set-language-environment 'Chinese-GB)
(setq gnus-default-charset 'chinese-iso-8bit
  gnus-group-name-charset-group-alist '((".*" . chinese-iso-8bit))
  gnus-summary-show-article-charset-alist
      '((1 . chinese-iso-8bit)
        (2 . gbk)
        (3 . big5)
        (4 . utf-8))
  gnus-newsgroup-ignored-charsets
      '(unknown-8bit x-unknown iso-8859-1))
;如果邮件里有 html 代码，则只显示其中的文字
 (eval-after-load "mm-decode"
  '(progn
     (add-to-list 'mm-discouraged-alternatives "text/html")
     (add-to-list 'mm-discouraged-alternatives "text/richtext")))
```

2．接收邮件

由于本节介绍的方法是在 Emacs 中接收邮件，所以应该先运行 Emacs。选择 Emacs 的菜单“Tools”|“Read Net News（Gnus）”命令，就开始了邮件的接收工作。等待一小段时间后，Emacs 就会将接收到的邮件展示在用户面前。Emacs 的“收件箱”如图 12.20 所示。

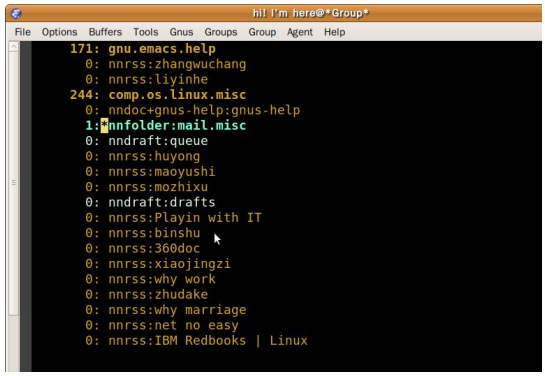


图 12.20　Emacs 的收件箱

在图 12.20 中，“1:*nnfolder:mail.misc”项代表邮件的收件箱。有“nnrss”前缀的是 RSS 源，而没有任何前缀的则是新闻组。单击 1:*nnfolder:mail.misc 选项，就进入了邮件浏览窗口，如图 12.21 所示。

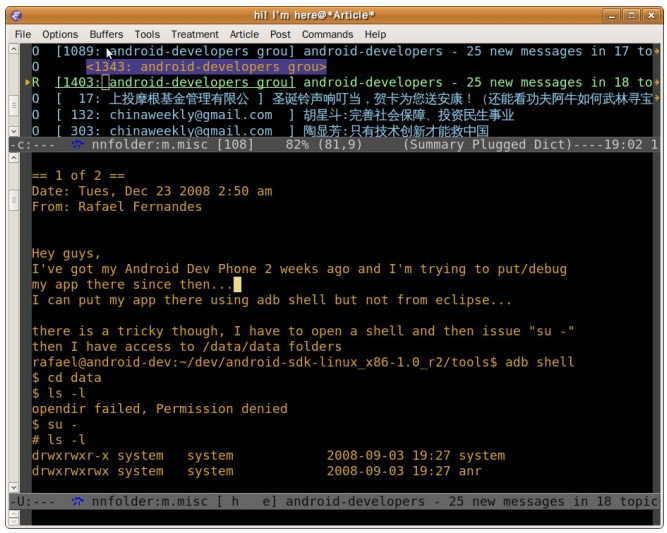


图 12.21　邮件浏览窗口

进入邮件管理窗口后，按空格键就会自动翻页显示邮件的内容。如果一封邮件显示完毕，按空格键会显示下一封未读邮件，这个功能非常实用、方便。

如果要回复正在阅读的邮件，简单地按 R 键即可。其他更多与邮件相关的选项，如转发、全部回复等，可以查看图 12.21 所示界面中的“Post”菜单获得。

3．发送邮件

在图 12.21 所示界面中按 M 键，即可开始创建新邮件。按下 M 键后，界面如图 12.22 所示。

在“To”中填写收件人的电子邮件地址，在“Subject”中填写邮件主题，然后在“--text follows this line—”下面开始写邮件正文。写完后按组合键“Ctrl+C Ctrl+C”，也就是按两次 Ctrl+C 组合键就可以将邮件发送出去。邮件写的过程中，要取消这个过程，可以按下“Ctrl+C，Ctrl+K”组合键，也就是先按下 Ctrl+C 组合键，再按下 Ctrl+K 组合键即可。

同样，在 Emacs 中，也可以发送带有附件的邮件。方法是在图 12.22 所示界面中选择 Attachments|Attach file 命令，然后定位到要添加的文件，就完成了向邮件添加附件的过程。

图 12.22 所示就是一封带附件的邮件。<#part 指示了附件的存在，而 type 指示了附件的类型是格式为 png 的图像，后面还有图片的位置等信息。

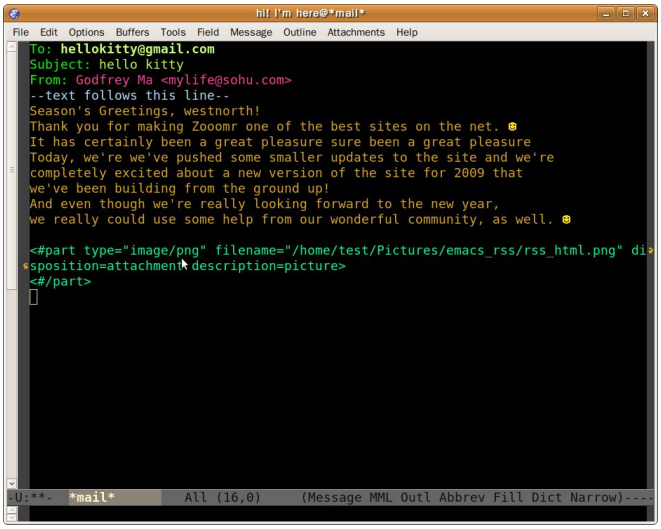


图 12.22　新建邮件



在 Emacs 中收发邮件的理由是其文本编辑功能较强。对于熟练的 Emacs 使用者来说，在 Emacs 中工作可以极大地提高编辑效率。而且 Emacs 不仅可以收发邮件，还可以阅读新闻组、阅读 RSS 源等内容，这样就不用打开多个软件，比较简洁。

12.6　即时通信软件

据调查，在我国使用即时通信软件（也就是聊天工具，如 QQ，MSN 等）的用户超过电子邮件，成为最重要的互联网应用，这使得 QQ 成为最大的互联网公司。所以，Linux 如果要推广，那么即时通信软件是极重要的一环。本节介绍的聊天软件，虽然并未完全兼容 QQ（如无语音、无视频），但功能却毫不比其他任何即时通信软件逊色。

12.6.1　全能的 Pidgin

Pidgin 是 Linux 平台中一款很强大的多功能即时通信软件。而且其是多平台支持的，也有 Windows 版本。其支持 QQ、MSN、GTalk 流行聊天软件。

1．软件界面及特点

Pidgin 软件在 Ubuntu 发行版中已经附带，所以不用安装，直接就可体验其强大的功能了。

单击主菜单“应用程序”|“互联网”|“Pidgin 互联网通信程序”命令，就可以打开该程序了。

在还没有开始使用之前，先看看其界面，界面如图 12.23 所示。

其主要优点如下。

- 集成多个即时通信软件，用一个软件即可同时登录多个即时通信软件。
- 速度快，占用内存少。由于其没有其他臃肿的附带功能，所以软件本身很小巧。
- 界面简洁。一般来说，就只有一个发送聊天内容的窗口。
- 插件比较多，有的是软件自带，也有的是热心网友开发的。这些插件可以支持其他软件，如安装飞信的插件后就可使用飞信发送免费的短消息等。

虽然其具有如上的优点，但对于用户而言，其缺点也是显而易见的。

- 功能不如原生的软件丰富，如缺少相应音视频功能等。
- 不太稳定，有的内容可能显示错误，如 QQ 群组名。

不过，客观地说，上述缺点其实责任不在 Pidgin，而是原生的软件没有开源或没有遵守相关的开放互联协议造成的。



图 12.23　Pidgin 的界面

2．添加账号

Pidgin 默认是没有账号的，所以运行程序后第一件事就是添加账号。选择“账号”|“管理”命令，在弹出的对话框中单击“添加”按钮，出现如图 12.24 所示的对话框。本例以 MSN 账号为例，介绍添加账号的方法，其具体步骤如下。

（1）在图 12.24 所示对话框“协议”下拉列表框中选择 MSN 选项，意思是添加的账号使用 MSN 协议。

（2）“屏幕名称”文本框实际上是指账号，应该在其中输入 MSN 账号信息。

（3）密码就不用说了，很明显应该输入该账号的密码。

（4）“本地别名”是指显示在 Pidgin 中的名字，取一个可以和其他账号有明显区别的名字就行了。

（5）如果选择了“新邮件通知”复选框，如果此账号对应的邮箱收到新邮件，就会在 Pidgin 中弹出提示，可参见图 12.23。

添加 QQ 也可以使用类似的方法，在第 1 步要将协议更改为 QQ，同时“屏幕名称”文本框中填入 QQ 号码。其他内容与添加 MSN 类似，这里就不再赘述。

如果 QQ 登录失败，可以选择图 12.24 所示对话框中“高级”标签，打开高级选项卡，如图 12.25 所示。将“服务器”文本框内容设置为“219.133.60.173”，将“端口”文本框内容设置为 80。这样 QQ 的登录问题就应该可以解决了。



图 12.24　添加账号



图 12.25　QQ 高级设置

Gtalk 在端口被封的情况下，也可能会出现登录错误的情况。这时在图 12.24 所示对话框中单击“高级”标签，在高级设置标签页中选中“强制旧 SSL”复选框，将连接端口设置为 443（每个协议的高级标签页中的内容都不相同，所以 Gtalk 的高级选项与图 12.25 显示的内容不同）。

MSN 端口被封的情况下，可以在“高级”选项卡中设置“使用 HTTP 方式”。

12.6.2　可拨打普通电话的 Skype

Skype 也是一款即时通信软件。对于 Linux 来说，其提供了原生的音、视频功能，所以显得难能可贵。如果购买了 Skype 的 SkypeOut 卡，还可以拨打国际长途电话，资费比较便宜。打到美国及世界大多数国家的资费标准均是 0.187 元人民币/分钟。

Skype 采用了先进的 P2P 技术，可以提供超清晰的语音通话效果；而且使用了端对端的加密技术，可以保证通信的安全；在局域网中也可轻松使用。具体来说，Skype 的优点如下所示。

- 语音质量极高：Skype 提供了高质量的语音通信，其声音质量高于其他聊天软件，甚至也高于普通电话的通话质量。
- 可轻松穿过防火墙：在其他通信软件无法正常使用的场合均可正常使用 Skype，这在一些局域网中比较实用。
- 可实现多方通话：Skype 可进行最多 5 方通话。而且通话过程是全程加密的，保密性强。
- 传送大文件：可以在加密通道中传送大文件，文件大小可达数 GB，可以跨平台传送。
- 全球共享：可以查找添加全球的联系人进行添加。
- 安全性高：采用顶级加密技术，传送文件、发送短消息、语音通话均采用加密传送，保证安全。
- 跨平台性：可在 Linux、Mac、Windows 各平台使用。

1．安装

安装很简单，使用“sudo apt-get install skype”即可。安装后，在菜单中选择“应用程序”|“互联网”|“Skype”命令即可打开。登录界面如图 12.26 所示。

2．登录

如果没有账号，则单击“还没有 Skype 账号吗？”超链接（如图 12.26 所示），参照网页上的说明，在 Skype 网站上注册一个 Skype 账号。然后将账号和密码输入相应的文本框，单击“登录”按钮进行登录，登录成功后，即进入如图 12.27 所示界面。



图 12.26　Skype 登录界面



图 12.27　Skype 程序界面

3．拨打普通电话

如果有 SkypeOut，单击“拨打普通电话”按钮，弹出如图 12.27 所示数字键盘，输入电话号码即可拨打电话。但如果没有 SkypeOut，则无法呼叫成功。现在国内用户也可以购买 SkypeOut 了，地址是 http://skype.tom.com/products/redeem/redeem_pro.html?card_type=out。

4．开启音、视频聊天功能

Linux 版的 Skype 支持音、视频聊天。如果 Skype 没有正确识别系统的音、视频硬件，可能会无法使用音频、视频功能。不过，这也是很容易解决的，参照如下步骤进行设置一般可以解决问题。

（1）单击 Skype 界面中左下角的 Skype 图标，在弹出菜单中选择 Option（选项）命令，出现如图 12.28 所示对话框。

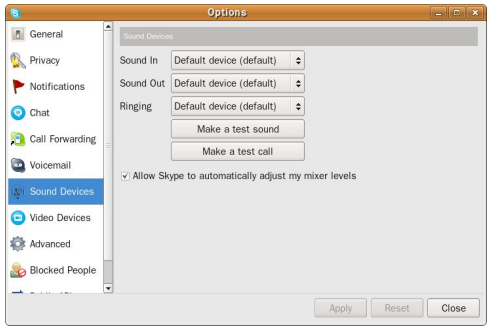


图 12.28　设置音频、视频设备

（2）单击左侧列表框的“Sound Devices”（声音设备）进行音频硬件的设置。

（3）分别在 Sound In（音频输入）、Sound Out（音频输出）、Ringing（响铃）下拉列表框中选择适当的设置。如果不知道自己的音频设备是哪种，可以逐个选择音频硬件下拉框中的项进行试。

（4）单击 Make a test sound（测试音频输出）与 Make a test call（拨打测试电话）按钮，如果硬件设置没有问题，应该可以听到声音。

（5）如果听不到声音，回到第 3 步，更改硬件设置，再进行测试。

（6）单击左侧列表框的 Video Devices（视频设备）进行视频硬件的设置。这个设置很简单，Skype 会提供一个向导，用来设置视频设备。这里就不再详细说明了。

5．添加联系人

如果知道某人的 Skype 账号，就可以在 Skype 界面（图 12.27）左下方 Skype 图标旁边寻找绿色加号，然后单击该图标。在最上方的文本框中输入要查找的 Skype 账号信息，单击“搜索”按钮开始搜索，很快会搜索出相关的结果，单击要添加的用户，然后单击“添加联系人”按钮即可将其添加至联系人列表。

经过如上的设置，用户应该就可以享受 Skype 的高质量音频通话和视频聊天了。



现在许多手机都提供了 wifi 上网功能，而且在许多手机里也可以安装 Skype。在手机里安装一个 Skype，然后来到提供免费 wifi 上网的场所，就可以免费与其他 Skype 用户聊天了。

12.6.3　专业强大的 IRC

IRC 是 INTERNET RELAY CHAT 的缩写，意思是互联网中继聊天，其通过特殊的协议（RFC1459 IRC 协议）使得许多用户连接到一台或多台 IRC 服务器上进行聊天。IRC 最主要的特点是功能多，提供了数以百计的命令来进行控制，可以实现许多独具特色的控制功能，这些强大的功能是 QQ 之类纯娱乐性质的聊天软件所无法比拟的。

IRC 有服务器和频道的概念。服务器是提供 IRC 服务的主机的地址，而频道是一个相关话题的集中讨论处，如#ubuntu 频道一般均讨论 Ubuntu 相关的内容。仔细查看网页，就可以发现国外许多技术站点均有一个名为 IRC 的页面。而作为一款优秀的即时通信工具，IRC 在国内很少见到其踪影，不能不说有点遗憾。

1．在 Pidgin 中添加 IRC 账号

IRC 有许多客户端，如 Opera 等软件均有 IRC 功能，但还是推荐使用万能聊天工具 Pidgin 中的 IRC 功能。依次选择“应用程序”|“互联网”|“Pidgin 互联网通信工具”命令打开 Pidgin，然后选择“账户”|“菜单”|“管理”命令，在弹出的对话框中单击“添加”按钮，出现图 12.29 所示对话框。



图 12.29　添加 IRC 账户

在“协议”下拉列表框中选择 IRC 选项。屏幕名称就是 IRC 的账户，如果有，就填上去，没有的话可以先随便填一个。服务器可以使用默认的“irc.ubuntu.com”，也可以使用“irc.freenode.net”。高级设置中，端口 6667，编码 UTF-8，均不用更改。单击“保存”按钮完成设置。完成以上步骤就已经连接到 IRC 服务器了。

2．IRC 使用简介

连接到 IRC 服务器后，Pidgin 会弹出一个如图 12.30 所示的对话框。其界面主要由两个窗口组成，一个窗口用来显示命令及聊天内容，其下的窗口用来输入命令及聊天内容。



图 12.30　登录到 IRC 服务器的提示

在命令及聊天内容输入窗口中输入“/join #ubuntu-cn”，然后按下 Enter 键。以“/”开始的输入被视为 IRC 的命令，本次输入命令的意思是加入 Ubuntu 的大本营 Ubuntu-cn 频道。这些命令后文会做较为详细的介绍。命令输入正确后，会出现几个窗口，注意图 12.30 中第二个标签页，标题是“#ubun…”，这就是刚才所加的 Ubuntu-cn 频道。单击该标签，进入该页，这才是 IRC 真正的界面，如图 12.31 所示。



图 12.31　IRC 界面

现在就可以做一名匿名者在该频道发言了。为什么说是匿名者呢？因为当前所用的用户名都是暂时的，以图 12.29 所输入的 Godfrey 为例，如果在 IRC 服务器上一个用户注册了 Godfrey 的用户名，而该用户现在如果登录，临时的“Godfrey”使用者将被踢出服务器。所以，应该尽快在服务器上注册一个属于自己的名字，以后就可以堂堂正正地用自己的名字登录 IRC 服务器了。

3．注册 IRC 用户名

注册用户名的方法很简单，向服务器发送“/nick 用户名”命令即可，如在图 12.30 下方的文本框中输入“/nick Godfrey”。如果出现“您现在叫做 Godfrey”，则说明该名字无人使用，注册成功。然后再用“/msg nickserv register 密码 电子邮件地址”来正式向服务器注册这个昵称，密码是“密码”的内容。如“/msg nickserv register 123456 mymail@sohu.com”，向服务器注册刚才申请的昵称，密码是 123456，而电子邮件地址是 mymail@sohu.com。

4．将常用频道加入 Pidgin 的好友列表

如果有常用的频道，可以将其加入好友列表，方便下次访问。方法是在图 12.31 所示对话框中选择“对话”|“添加”命令，出现如图 12.32 所示对话框。



图 12.32　将 IRC 频道加入好友

可以看到，当前频道已经显示在“频道”文本框中，在“别名”中输入一个名字，单击“添加”按钮就可将该频道添加到好友列表中。如果选择了 Autojoin when account becomes online（当账号在线时自动加入该频道）复选框，则当 Pidgin 启动时，如果有网络连接，会自动加入到该频道。如果没有选择该项，则 Pidgin 不自动加入该频道。

5．IRC 常用命令

当然，IRC 的命令还有许多，下面列出一些以供使用时参考，要记得在命令前加上“/”，不然 IRC 会将这些命令也当做普通文本发送的。

- away：使用/away “I anned”，显示“I anned”消息，暂时离开。
- help：显示帮助信息。
- invite 用户名 频道：邀请某人到某频道。
- join：加入某个频道。
- kick 频道 用户名：将用户从某个频道踢出去。
- list：显示当前服务器所有频道。

- me：做动作让别人看到。
- nick：使用一个昵称，这个昵称可以任意选择。
- whois：查询某个用户的信息，如 whois zhangsan。
- ping：获得网络速度。
- say：与直接输入文字效果一样，主要用于脚本。
- time：获得当前服务器时间。
- notice：向某人发送消息，如/notice zhangsan “hello”。
- part：退出某频道。
- query：向某人发送聊天信息。
- quit：退出连接。

12.7　小结

本章介绍了 Linux 中网络相关软件，如网络浏览器 Firefox、电子邮件管理软件、即时通信软件等，除了常见的软件外，本章还介绍了相关的一些字符界面程序，这些程序功能也非常强大，而且速度更快，在强调速度的场合有其用武之地。本章的重点内容是 Firefox 及 Pidgin 的使用，因为网络浏览与聊天是大多数用户上网的重要内容。而本章的难点就是相应的字符界面软件的使用。因为这些软件一般是依靠键盘操作，所以要求用户记忆许多操作键。不过多加实践后会感觉到字符界面程序还是有其优势的。

第 13 章　多媒体软件

多媒体应用虽然并不是 Linux 的强项，但凭借开源共享的精神，Linux 下汇集了一大批优秀的开发者，在多媒体软件方面还是取得了很大的进展。现在 Linux 系统中已经拥有了许多功能强大、界面美观的视频、音频播放器，以及优秀的视频处理软件，可以完成专业的视频处理工作。本章将介绍这样一些优秀的软件。

13.1　视频播放

许多人在辛苦的工作之余，会在计算机上看看电影、听听歌来缓解工作的压力。Ubuntu 虽然安装了许多播放软件，但由于版权的问题，并没有提供最基本的解码器（如最基本的 avi 电影、现在很流行的 rmvb 格式视频等）。这就需要安装很多的解码器来充实这些播放软件，使之可以播放出美丽的音乐和动人的电影。本节将简单介绍解码器的安装和几款常用的播放器的使用。

13.1.1　全能电影播放器 MPlayer

在 Linux 领域，MPlayer 是当之无愧的视频播放王者。其功能极其强大，支持几乎所有的影音格式文件、支持许多网络视频协议、可以播放网络视频（这个特点被用来当作网络电视）、可以播放 DVD 碟片、可以收听网络电台等。而且其支持目前所有的主流个人电脑平台，如 Windows、UNIX、类 UNIX 系统，以及 Mac OS 等。

但是，由于版权的原因（mp3 是私有版权，rmvb 更是专利版权），MPlaye 附带的解码器支持格式很少。而且 MPlayer 是基于命令行的，对于普通用户来说使用可能还是有一定困难的。所以，在 MPlayer 正常使用前，需要先安装一些解码器。为了美观，也可安装一些软件 skin（皮肤），下面是 MPlayer 的一些设置方法与功能简介。

1．安装解码器，以支持更多的格式

安装解码器的主要目的是为了正确地让播放软件来进行音频、视频的解码，然后正常播放。而添加了一些 Ubuntu 的镜像服务器后，安装解码器也就成为一项很简单的工作。可以简单通过 Ubuntu 的 apt 包管理器或“新立得软件管理器”进行安装。

（1）首先添加解码器所在的源，添加源后方可使用 apt 安装。方法是在终端中运行如下命令：

```
sudo wget http://www.medibuntu.org/sources.list.d/hardy.list -O /etc/apt/sources.list.d/medibuntu.list
```

（2）刷新软件源，并安装 MediUbuntu 的密钥。

```
sudo apt-get update && sudo apt-get install medibuntu-keyring && sudo apt-get update
```

（3）再安装多媒体解码器，运行如下命令：

```
sudo apt-get install libxine1-ffmpeg libxine1-all-plugins libxine1-plugins
```

```
w32codecs gcc-3.3-base libstdc++5
    #Xine 多媒体引擎解码器
    sudo apt-get install gstreamer0.10-ffmpeg gstreamer0.10-pitfdll gstreamer0.
10-plugins-bad  gstreamer0.10-plugins-bad-multiverse  gstreamer0.10-plugins-
ugly gstreamer0.10-plugins-ugly-multiverse gstreamer0.10-esd
    #Gstreamer 多媒体引擎解码器
    sudo apt-get install w32codecs
    #最好再安装这个解码器包，就可以正常播放所有 windows 中常用的音、视频格式文件。
```

（4）安装 DVD 影碟功能支持，使用如下命令：

```
    sudo apt-get install libdvdnav4 libdvdread3 libdvdcss2
```

（5）安装 mplayer 及其图形前端，使用如下命令：

```
    sudo apt-get remove totem-mozilla -y        #卸载 Totem 的 firefox 插件
    sudo apt-get install mplayer mozilla-mplayer mplayer-fonts mplayer-skin
mplayer-skins mplayer-skin-blue
    #安装 mplayer、mplayer 的 firefox 插件、以及 mplayer 的外观
```

2．软件外观

安装了 mplayer-skin-blue 的 MPlayer 界面如图 13.1 所示。

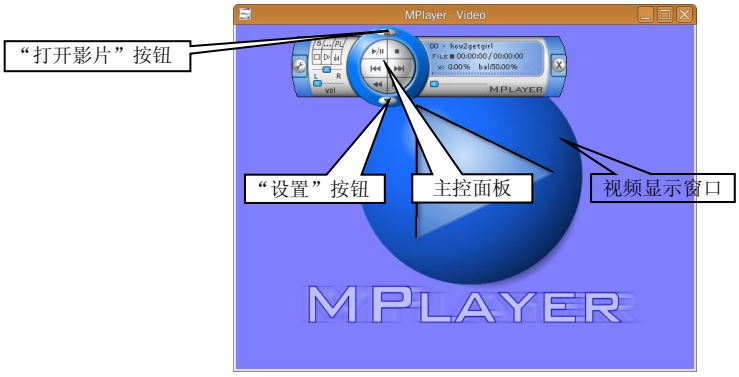


图 13.1　MPlayer 界面

图 13.1 所示的“主控面板”与“视频播放窗口”是可分离的两个部分。“主控面板”用于打开视频，调整音量、拖动进度条、及进行系统设置。“视频播放窗口”显示正在播放的视频内容。

3．加载字幕

用户可以使用“打开影片”按钮打开视频文件进行播放。如果该视频文件还有字幕文件，加载字幕的方法也很简单，直接将字幕拖入“视频播放窗口”即可。

有时候“视频播放窗口”中显示的中文字幕是乱码，这说明没有合适的中文字体用于显示字幕。可以参照如下标准进行设置。

（1）单击“主控面板”中的“设置”按钮或者在“视频播放窗口”中右击，在弹出的菜单中选择 Preferences（首选项）命令，出现如图 13.2 所示的对话框。

（2）单击 Font 文本框后的 Browse 按钮，找到合适的中文字体，如 wqy-zenhei.ttf。这是用来显示字幕的字体，一定要是中文字体。

（3）将 Encoding 选项设置为 Unicode。

（4）拉动 Text scale 后的滑动块，将其设置为合适的比例，如 2.05，这是控制字幕大小的。回到视频中，如果发现字幕太大，则缩小该比例，反之，则扩大该比例。经过这些设置后，中文字幕文件应该可以正常显示在视频中了。

4．打开视频文件错误的处理方法

如果打开视频文件时，提示找不到指定的设备，也可用如上所示的方法打开 Preferences 对话框，参照如下步骤进行相关的设置。

（1）选择 Video 标签，打开视频设置标签页，如图 13.3 所示。

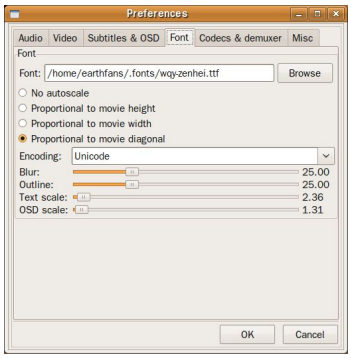


图 13.2　字幕中文显示设置

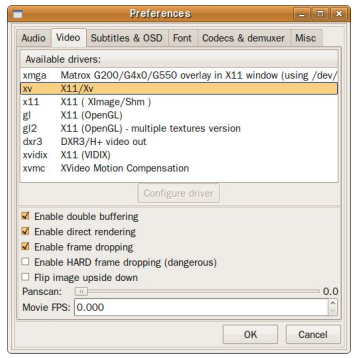


图 13.3　视频设备设置

（2）在 Available drivers 列表框中选择适当的设备，最常用的是 xv，也可以试用其他选项，以找到最适合自己机器的选项。

（3）选择 Enable double buffering（启用双缓冲）和 Enable frame dropping（启动失帧识别）功能，可改善某些无法播放的情况。



如果音频输出出现问题，也可参照如上步骤进行设置。方法是在图 13.3 所示对话框中选择 Audio 标签，打开音频设置对话框进行音频输出的设置。

5．MPlayer 的快捷键

MPlayer 一个很主要的功能是组合键，在播放中使用组合键可以方便许多。比如，许多用户在开始使用 MPlayer 时，会发现无法拖动进度条来控制播放进度。而其实 MPlayer 提供了方向键来控制进度的方法，可以更为方便地调整进度。下面是一些 MPlayer 可用的组合键列表。

- 上方向键：后退 1 分钟。
- 下方向键：前进 1 分钟。
- 左方向键：后退 10 秒。
- 右方向键：前进 10 秒。
- PgUp：后退 10 分钟。
- PgDn：前进 10 分钟。
- f：进入全屏状态。在全屏状态下按下 F 键则退出全屏，回到普通显示状态。
- 空格：可将影片暂停，按任意键，就会开始播放。
- m：静音。
- < 与 >：播放列表中上一部或下一部节目。

- q 或 Esc：退出。
- ＋ 和 －：有些电影中的音频与视频会发生不同步的情况，如演员的嘴动了半天才听到了声音，这时就要调整音频延迟。每按“＋”一次，音频向后调整 0.1 秒，“－”与相反。
- “/ 和 *”或者“9 和 0”：降低和提高音量。 / 和 9 是降低音量，*与 0 提高音量。
- o：显示播放时间信息，如已播放时间等。
- v：显示或隐藏字幕。
- z 和 x：如果字幕与影片出现不同步的情况，就用这个调整。单位是 0.1 秒。
- r 和 t：调整字幕位置。
- 1 和 2：降低和提高对比度。
- 3 和 4：降低和提高亮度。
- 5 和 6：降低和提高色度。
- 7 和 8：降低和提高饱和度。

13.1.2　电影播放器 Totem

Totem 是 Ubuntu 系统自带的视频播放器，界面比较简单。在安装了 13.1.1 节所示的解码器后，Totem 应该可以正常播放绝大多数格式的视频文件。Totem 与其他 Linux 平台软件一样，支持插件。下面介绍一些关于插件及字幕加载的相关知识。

1．Totem 的 YouTube 插件

Totem 内置有许多插件。其中有一款插件很有趣，可以直接播放 YouTube 中的视频。启用了 YouTube 插件的 Totem 界面如图 13.4 所示。

启用该插件的方法如下。

（1）依次选择 Totem 菜单的“编辑”|“插件”命令，打开如图 13.5 所示对话框。



图 13.4　Totem 界面与 YouTube 播放插件



图 13.5　Totem 插件配置对话框

（2）选择“YouTube 浏览器”复选框，意思是启用该插件。

（3）在图 13.4 所示下拉框中选择 YouTube，即可出现 YouTube 搜索文本框。

启用了该插件后，如果计算机与互联网连接，就可直接观看 YouTube 的相关视频了。方法如下所示。

（1）在图 13.4 所示右侧的文本框中输入要观看视频的关键字（如 Wall E），该插件会调用

YouTube 的 API 搜索与该关键字相关的所有视频。

（2）单击搜索按钮（如图 13.4 所示），开始搜索。

（3）稍等一会，搜索结果会显示在右侧列表框中。这些搜索结果会以与搜索关键字的相关程度排序，一般会将最相关的内容显示在第一条。

（3）双击搜索结果中的任一条目，则该视频会在 Totem 的视频播放窗口中开始播放。



在许多地区，由于网速原因，播放视频时会出现停顿现象，这是很正常的。只要单击暂停按钮等待一会儿，然后播放就会比较正常了。

2．字幕加载

随着与世界交流的增多，许多计算机用户经常会观看一些外语视频节目。但是许多人的外语水平可能还不足以完全理解节目内容。这就需要加载字幕进行观看。Totem 提供了加载字幕的功能。这个功能的实现比 MPlayer 稍微复杂些，其步骤如下。



图 13.6　Totem 首选项对话框

（1）选择 Totem 菜单的“编辑”|“首选项”命令，打开如图 13.6 所示对话框。

（2）选择“常规”标签，打开“常规”选项卡。

（3）在“常规”选项卡中选择“加载视频时自动加载字幕文件”复选框。

（4）在“字体”下拉列表框选择“文泉驿正黑 中等”字体。当然，选择其他中文字体也可。

（5）将“编码”下拉列表框内容设置为“简体中文（GB18030）”，也可设置为 UTF-8。

经过如上设置，一般而言，Totem 会自动加载与视频同名的字幕文件，将其显示在视频中。

如果中文字幕显示乱码，可以参照如下步骤编译 Totem 的中文字体，以使其正常显示中文字幕。

（1）使用任何文本编辑器以 root 用户身份打开源列表，如“sudo vim /etc/apt/sources.list”。在文件的最后增加一句，内容如下：

```
    deb-src http://archive.ubuntu.com/ubuntu/ breezy main restricted universe
multiverse
```

这是添加了包含了源代码的源服务器。

（2）下载源代码，编译 Totem 字体生成程序 xine-fontconv。

```
    sudo apt-get update                    #刷新源列表
    sudo apt-get source libxine-dev        #下载 libxine-dev 的源代码
    cd xine-lib-1.1.11.1/misc/             #进入刚下载的源代码的 misc 目录
    sudo gcc -o /usr/local/bin/xine-fontconv xine-fontconv.c `freetype-config
--cflags --libs`
    #编译 xine-fontconv.c，生成 xine-fontconv。
```

（3）使用上步编译的 xine-fontconv 程序，可以普通字体为基础，生成 Totem 字体，命令如下：

```
    cd /usr/share/xine/libxine1/fonts        #更改当前目录到 Totem 字体目录
    sudo xine-fontconv ～/.fonts/wqy-zenhei.ttf zeihei gb18030 #使用前面编译的
```

程序生成 Totem 字体。

前面命令中使用的字体文件“～/fonts/wqy-zenhei.ttf”必须是已经存在的一个字体，该项可由用户依据情况自行修改。

（4）打开“～/.gnome2/totem_config”文件，将字体文件的相关设置修改为如下内容。

```
subtitles.separate.font:zeihei
subtitles.separate.src_encoding:gb18030
# tiny small normal large very large huge
subtitles.separate.subtitle_size:large
#字体一共有六个大小，个人感觉用 large 比较理想
#如果用户的屏幕适合其他的，也可以将其设置为上述其他值
subtitles.separate.vertical_offset:40        #这是字幕距底端的距离
```

Totem 不能正常显示中文字幕，是因为其没有合适的中文字体。而且有些版本的 Totem 无法使用普通的字体文件，必须使用特定的程序从现有的字体基础上生成 Totem 所能识别的字体。而这个用以生成 Totem 字体的程序只提供了源代码，所以不得不下载源码自己编译。这就是上述步骤所做的内容。

现在打开电影，如果电影文件所在目录中有与电影同名的字幕文件，就会在 Totem 中正常显示。如播放的电影名是 abc.avi，而该目录中有一个名为 abc.srt 字幕文件，用 Totem 打开就可以正常显示字幕了。

13.2　音频播放

音乐可以慰藉人类的心灵，在进行紧张的工作的同时，计算机中传来优美的音乐，无疑会使人心情愉悦。Linux 中拥有许多优美、强大的音频播放软件。掌握这些软件的使用方法，无疑会给生活增添许多乐趣。本节主要介绍 Audacious 与 Rhythmbox 的使用。

13.2.1　类似 Winamp 的播放器 Audacious

Audacious 是 gtk 平台的一款外观漂亮，功能强大的音乐播放器。Audacious 可以直接使用 Winamp 2.x 的皮肤。也就是说可以有基本与 Winamp 一样的用户体验，这对于习惯了 Winamp 的用户来说是个福音。Audacious 自带了一些解码器，默认情况已可以支持 mp3、aac、wma 格式。其官方站点是 http://Audacious-media-player.org。

1．安装

Audacious 的安装很简单，只需在终端运行如下两条命令，安装主程序及相应的解码器即可。

```
sudo apt-get install audacious                          #安装 Audacious
sudo apt-get install audacious-plugins audacious-plugins-extra audacious-
plugins-extra-console
#安装 Audacious 的解码器
```

安装完成后界面如图 13.7 所示。

使用就更简单了，单击主界面的相关按钮即可完成相应的工作。基本操作与 Winamp 完全一样，相信对于普通用户来说使用很容易上手。



图 13.7　Audacious 主界面

2．安装歌词显示插件

有的用户喜欢在听歌时看歌词，原因有二：一是因为有时是想学英语（听英文歌曲时）；二是有时听不清中文歌词（听香港某流行天王的歌）。所以这些用户需要一款能在播放音乐的同时显示歌词的插件。南京 Allan 的 Showlyric 插件就可以解决这个问题，插件的功能如下：

- 歌词自动下载；
- 支持加载将歌词保存到本地并从本地加载歌词；
- 歌词窗口总在最前并具有透明效果，智能显示歌词窗口（当无法找到歌词时自动隐藏）；
- 配置歌词保存路径；
- 智能适应歌词宽度；
- 选择歌词下载地址或自动下载；
- 从千千静听的服务器下载歌词；
- 编辑歌词功能；
- 鼠标调整时间功能；
- 设置歌词字体。

安装并设置了该插件后，Audacious 的界面会类似于图 13.8 所示。很漂亮吧，按照下面的步骤，读者也可以将 Audacious 打扮得如此美丽。

（1）在如下地址下载该插件 http://forum.ubuntu.org.cn/download/file.php?id=38389，得到文件 showlyric.tar.gz。

（2）将插件文件解压，得到 Showlyric 目录，从终端中来到该目录，如“cd Showlyric”。

（3）在终端中执行“sudo ./install”，就可完成安装。

（4）按下 Ctrl+P 组合键打开首选项菜单，如图 13.9 所示。



图 13.8　歌词显示插件效果

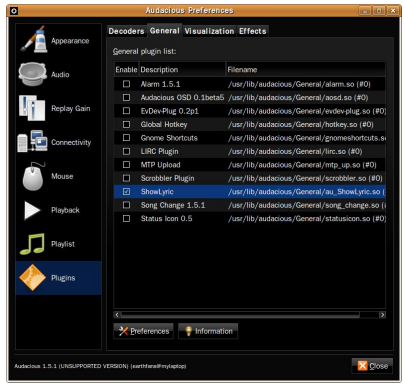


图 13.9　歌词显示插件设置

（5）选择左侧列表中的 Plugins 选项，打开插件页。

（6）在插件页中打开 General 选项卡，在其中选择 ShowLyric 复选框。意思是打开插件 ShowLyric。

（7）回到主界面，保证连接到网络，单击任一首歌。Audacious 就会去网站查询歌词，然后显示在歌词窗口中。Audacious 还会将歌词保存在本地，以便下次显示。

如果对于歌词显示的颜色不满意，就在歌词窗口中右击，会弹出歌词设置对话框，可以在其中进行设置。

3．设置默认音量大小

Audacious 默认的音量要小于系统设置的音量。可以将其设置为与系统同样大小，具体步骤如下：

（1）在 Audacious 中按下 Ctrl+P 组合键打开 Preferences（首选项）对话框；

（2）选择左侧 Replay Gain 的选项，找到 Defult replay 微调框，将其设置为 0，意思是默认与系统同样的音量。

4．文件乱码的解决方法

在打开文件时，有时会出现文件名乱码的情况，这是由于没有正确设置文件编码，按如下步骤设置文件编码后就可解决该问题。

（1）按下 Ctrl+P 组合键打开 Preferences 对话框，选择左侧的 Playlist 选项，如图 13.10 所示。

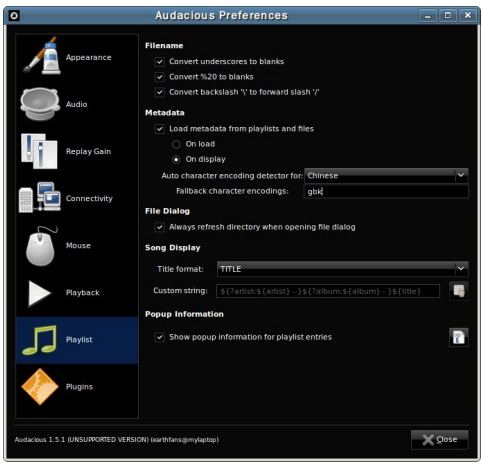


图 13.10　文件编码设置

（2）将其 Auto character encoding detector for（自动检测文件编码）下拉框框设置为 Chinese。

（3）在 Fallback character encoding（备选方案）文本框中输入 gbk。

（4）关闭设置项，重新打开文件，就会发现原来乱码的文件已经正常显示了。

13.2.2　类似 iTune 的播放器 Rhythmbox

Rhythmbox 是一款类似于 iTune 的音乐管理与播放软件，也就是说其不只是一款可以播放音乐的软件，同时具有强大的音乐文件组织与管理功能。其主运行界面如图 13.11 所示。

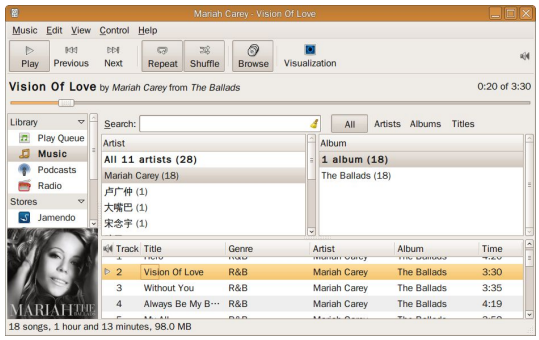


图 13.11　Rhythmbox

其是 Ubuntu 默认的音乐播放与管理软件，所以直接使用即可，无须进行安装。如果 Ubuntu 的某个发行版中没有安装，也可自行安装。安装的方法很简单，运行命令 sudo apt-get install rhythmbox 就可以了。安装后，在主菜单选择“应用程序”|“音频、视频”命令，打开“Rhythmbox 音乐播放器”即可。

在 Rhythmbox 中，打开某些 ID3 信息是中文编码的文件，在其界面显示的相关音乐信息时会出现乱码。不过，也不用担心，只要安装一个软件，执行一个命令就可以让 Rhythmbox 正常显示中文的信息了。下面是解决乱码问题的方法步骤。

（1）安装工具集，使用命令 sudo apt-get install python-mutagen。

（2）在终端中更改工作目录到保存音乐文件的目录，如使用命令“cd ～/Music”。

（3）执行命令“mid3iconv -e GBK --remove-v1 *.mp3”即可将当前目录中所有的以 mp3 为后缀的音乐文件中 GBK 编码的 mid3 信息转换为 Unicode 编码，并且在转换后删除 ID3v1 标签。

（4）如果要修改当前目录及当前目录中的子目录中的文件，也就是说要递归地修改音乐文件的标签信息，可执行此命令“find . -iname "*.mp3" -execdir mid3iconv -e gbk {} \;”，这条命令的意义可参阅第 6 章 Shell 基本命令的相关部分内容。

13.3　视频处理

随着 DV 与手机拍摄功能的增强，许多人的计算机里均保存着许多影像资料。但可能由于种种原因，这些录像并不是特别让人满意。如有时录入了不相干的内容、有时录像的内容显得有些单调等。这时就可使用 Linux 中强大的视频处理软件将这些视频进行剪辑、处理，使之达到理想的效果。

13.3.1　视频编辑软件 LIVES

LIVES 是 Linux 系统中一款功能强大的视频编辑软件，可以完成视频转换、视频剪切、给视频增加特效等任务。Ubuntu8.04 用户可以在这儿下载 Deb 包 http://www.getdeb.net/download/3235/0。下载得到的文件很小，只有 1MB 多。许多读者可能对文件的尺寸有所怀疑，其实软件的功能与其尺寸没有必然的联系。

在控制台中进入放有 LIVES 包的目录，用“dpkg -i 下载的 deb 包名”进行安装，很可能会提示依赖条件不能满足，就使用“sudo apt-get install -f”来安装所需的依赖包。安装依赖包后，再使用“dpkg -i *.deb”安装下载的 LIVES Deb 包就可以了。在主菜单中选择“应用程序”

|“音频、视频”|“LIVES”命令，或在终端中输入 lives 就打开该软件。

1．主界面

LIVES 的界面如图 13.12 所示。图示的是 LIVES 的一种柔焦效果，界面中其他部分的元素在这里做一个简要的介绍。



图 13.12　LIVES 界面

- 界面主要部分显示两帧图像，分别代表当前选中的视频的第一帧和最后一帧。
- 界面中下位置的刻度尺代表视频的时间长度。
- 两个长条分别代表视频文件中的视频流和音频流。可以用鼠标在长条上单击以选择视频的起始帧，选中的部分以白色显示，而没被选中的部分以黑色显示。
- 界面底端是状态行，会滚动显示当前操作的相关信息。
- 菜单执行主要的操作，在菜单的右端，还有播放当前视频的按钮。单击 Play 按钮，会在两帧图像中间打开一个窗口，在里面播放视频。

2．打开文件

打开视频有两种方式，直接打开和打开指定帧。在菜单 File 中选择 Open file/directory 命令，则直接打开文件。但有时可能需要打开文件的一部分帧，这种情况可以选择 Open file selection 菜单项，会提示你输入打开的范围，以帧为单位，如图 13.13 所示。



图 13.13　选择打开的范围

在 Select start time（选择起始时间）文本框中输入要从第多少秒打开的数字，默认是 0。在 Number of frames to open（打开的帧数）文本框中输入要打开的帧数。如果对于选择不是很明确，单击对话框下面的 Click here to Preview the video（单击这儿显示视频预览）按钮，就可以在对话框中显示视频文件以便选取。

3．剪切、复制视频片断

LIVES 中，如图 13.12 所示音、视频流中显示状态为白色的部分为选中，选中的视频片断可以执行复制、剪切等操作。打开的文件默认是全部选中的，试着用鼠标单击文件的前后各三分之一处，会选中中间三分之一的视频。

选择 Edit|Copy selection 命令，或者直接使用组合键 Ctrl+C，就将当前所选视频复制了一份。然后按下 Ctrl+N 组合键可将复制的视频片断粘贴在新的视频文件中。

选中的视频，除了可以完成复制功能外，还可以在菜单 File 中选择 Encode selection as 命令，将选中的片断导出为另一种视频格式。当然，还可以将选中的片断按下 Ctrl+T 快捷键剪切出来，或按下 Ctrl+D 快捷键将其删除，以去除视频中效果不好或需要去除的部分。

4．特效制作

LIVES 支持许多特效，如前面提到过的画面模糊功能等，这些功能的实现均在 Effects 菜单中。如 Charcoal 就是选择 Effects|Charcoal 命令，其余类似，这里就不再一一注明。

- adjust contrast：调整对比度。
- Black frames：将前面选中的帧填充为一幅单色图像。
- Blur：将画面模糊化，如图 13.12 所示。
- Charcoal：炭笔画风格，如图 13.14 所示。

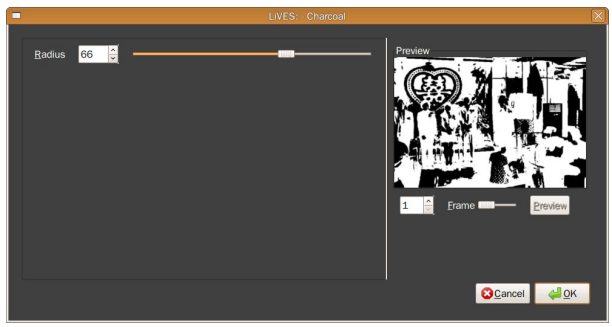


图 13.14　炭笔画风格

- Edge detect：描边风格，如图 13.15 所示。

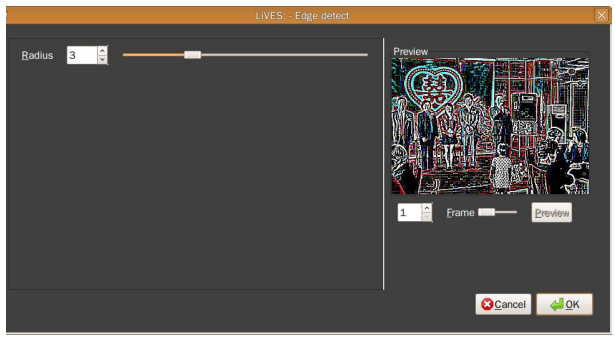


图 13.15　描边风格

- Emboss：浮雕效果。
- Enhance：优化图像。
- Monochome：图像黑白化。
- Flip vertically：垂直翻转。
- Flop horizontally：水平翻转。
- Spin：局部旋转部分图像。
- Shrink/Expand：图像由小变到大或由大变到小。
- Wave：让图像变成波浪效果。如图 13.16 所示。

- Midtone：设置中间色调。
- Swirl：让图像成为一个漩涡，如图 13.17 所示。

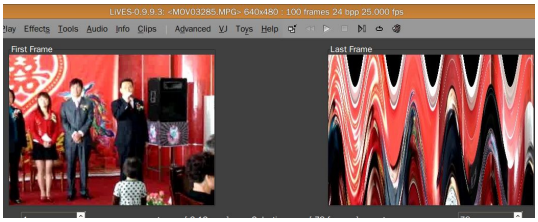


图 13.16　Wave 效果（左侧图像为原始素材）

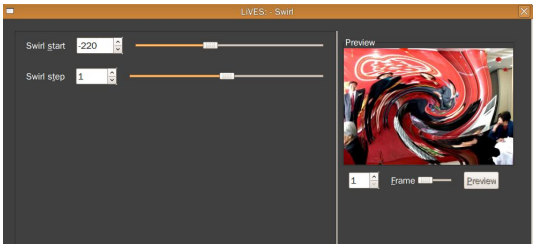


图 13.17　Swirl 效果

这些仍然是 LIVES 软件功能的一小部分，LIVES 还可更改视频大小、初步的处理音频、给影片增加音轨等功能。本节就不再详述了，有兴趣的读者请参阅相关专著。

13.3.2　视频格式转换 Avidemux

Avidemux 是 Linux 平台一款功能丰富且操作简单的视频格式转换软件，可以将 avi、vob 格式的视频转换为 DVD、VCD、SVCD、PSP mp4、IPOD mp4、FLV 等格式。还可以将图片转换为视频。该软件在 Ubuntu 的源列表中，因此安装是很简单的，直接在终端中输入命令“sudo apt-get install avidemux”执行就可以了。安装后的程序位于主菜单的“应用程序”|“音频视频”|“Avidemux”菜单项。或者直接在终端中输入 avidemux 就可打开该软件。一个典型的 Avidemux 界面如图 13.18 所示。

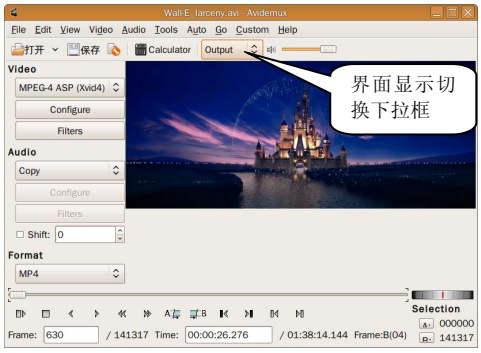


图 13.18　Avidemux 界面

1．简单的视频转换

要将一个 avi 或 DVD 碟中的视频文件转为其他格式，是很容易的。只要进行如下几步即可。

（1）单击工具栏“打开”按钮，在弹出的对话框中定位要转换的源视频文件，单击“确定”按钮打开该文件。有时可能 Avidemux 需要转换或提取视频中的某些信息，会给出提示对话框，单击“确定”按钮即可。

（2）打开后，视频文件的预览出现在界面的右侧。可以单击界面下端的播放控制按钮播放视频。

（3）在 Auto 菜单中选择要转换的格式，如 iPod(mpeg4)。

（4）在如图 13.18 所示的“界面显示”下拉列表框中，选择 Output 选项，指定 Avidemux 在界面中显示输出内容。

（5）在界面左侧可以看到内容已经变为输出的选项，“Video”格式是“MPEG-4 ASP”，意思是视频转换为该编码；“Audio”格式是“Copy”意思是复制源文件的音频格式，不转换；“Format”内容是“MP4”意思是输出为“MP4”格式。

（6）单击工具栏“保存”按钮，或按下 Ctrl+S 组合键，弹出保存文件对话框，在文件名文本框中输入转换后视频文件的全名，如“Wall-E.mp4”。单击对话框“保存”按钮即开始转换。

（7）转换结束后，会给出一个提示框，单击“确定”后，就在以查看转换后的视频文件了。

2．添加背景音乐

在视频处理中，经常会遇到需要给视频增加新的音轨的情况，如给一段视频加上音乐背景等。这在 Avidemux 也很容易实现。实现步骤如下所述。

（1）选择 Audio|Second Audio Track 命令，打开如图 13.19 所示的对话框。

（2）从 Audio source 下拉列表框中选择 External MP3（外部 MP3）选项。

（3）单击 External file 文本框后的 Browse 按钮，选择一个要作为背景的 MP3 文件。

（4）单击“确定”按钮就完成了音频文件的插入。



图 13.19　插入新的音轨

3．高级设置

第 1 条所介绍的视频转换方法选择了系统默认设置，但用户有可能存在着特定的需求，如设置自己的编码器、设置一些特效等。这些设置可以通过“Configure”按钮与“Filters”按钮来完成。具体步骤如下所示：

（1）先参照第 1 条“简单的视频转换”进行到第（5）步后，界面如图 13.20 所示。

（2）在“视频格式”下拉列表框中选择合适的格式，如 MPEG-4 ASP。

（3）单击“视频设置”按钮，打开如图 13.21 所示对话框。在其中可以设置编码类型，编码速率等各种与编码相关的选项。

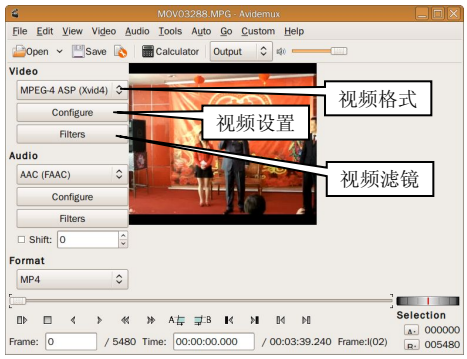


图 13.20　高级设置

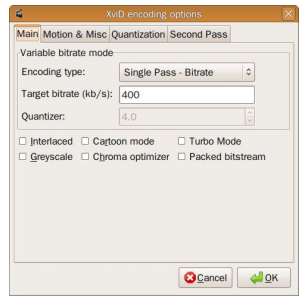


图 13.21　视频设置对话框

但是 Avidemux 的视频处理功能还远不至于此，使用“视频滤镜”还可以完成许多复杂的特效，如“切黑边”、对比度亮度的调整、提高画面质量等。其步骤如下所示。

（1）单击图 13.20 所示“视频滤镜”按钮，打开视频滤镜对话框，如图 13.22 所示。界面分为三部分，分别是“滤镜分类”、“滤镜名称”、“已启用滤镜”。

“滤镜分类”显示了当前系统中所有滤镜的分类，从名称上就可以看出其大概的功能。如 Transform 是变形方面的滤镜。Color 则是调整视频颜色相关的内容。而 Subtitles 提供了字幕加载方面的功能。Misc 里面的滤镜有“添加旋涡”以及给视频打 logo 等内容。



图 13.22　图像滤镜设置

（2）要添加“切黑边”特效，就选择“滤镜分类”里的 Transform 分类。

（3）“滤镜名称”中现在出现了一些与变形有关的滤镜，双击其中 Add black borders 选项，弹出对话框。

（4）按照要求在其中输入左、右、上、下的黑边的宽度，如“2，2，16，16”，单击 OK 按钮退出。

（5）现在特效已经加入。要查看效果，可以先看预览。方法是单击如图 13.22 所示的 Preview 按钮，会弹出一个视频窗口，拖动滑块就可以看到滤镜的效果了。如果不满意，还可以单击 Configure 按钮再重新设置。设置结束滤镜后，单击 Close 关闭对话框，完成设置。



LIVES 可以完成的特效基本上 Avidemux 均可以制作，而且后者还可进行视频格式的转换等额外操作。所以 Avidemux 是 Linux 系统中一款功能强大而实用的视频处理软件。

13.3.3　编译增强版的 FFmpeg

由于版权方面的原因，FFmpeg 在编译时没有将一些源码包含在其中。但这些源码对于转换某些格式的文件（如 iPhone 等设备可识别的特别格式的视频）是必需的。对于最终用户而言，可以将这部分源码包含在 FFmpeg 中，然后进行编译。编译成功后就可正常转换 iPod 等视频了。编译的方法其实并不难，简单地参照如下内容就可以完成了。

（1）获得编译 FFmpeg 所需的依赖文件，在终端中执行命令（其他步骤也同样在终端中执行命令）。

```
    sudo apt-get build-dep ffmpeg
    sudo apt-get install liblame-dev libfaad-dev libx264-dev libfaac-dev
libxvidcore4-dev liba52-0.7.4 liba52-0.7.4-dev libdts-dev checkinstall
```

（2）获得 FFmpeg 源码。

```
    apt-get source ffmpeg
```

（3）下载的源码位于“～/ffmpeg-0.cvs20070307”，注意后面的数字与版本号有关系。可以在输入“ffmpeg-”后按下 Tab 键，让 Shell 自动补全该数字。补全的结果一般就是正确的目录。

将当前目录更改为下载源码目录的命令如下：

```
cd  ～/ffmpeg-0.cvs20070307 #改变当前目录为下载的源码目录
```

（4）使用下列的命令进行编译前的设置工作：

```
./configure --enable-liba52 --disable-debug --enable-libfaad --enable-
libfaac --enable-gpl --enable-amr_nb --enable-amr_wb --enable-x264 --enable-
xvid --enable-libdts --enable-pthreads --enable-libvorbis --enable-pp –enable
-libtheora --enable-libogg --enable-libgsm --disable-debug --enable-shared
--prefix=/usr
```

（5）下载最新版本的 amr nb float，下载地址是：http://www.3gpp.org/ftp/Specs/latest/Rel-5/26_series/。



该页面中有许多版本，可以从其中选择最新的一个版本下载即可。

（6）将下载的压缩文件解压。

（7）将其中 c-code 目录中的所有文件复制到～/ffmpeg-0.cvs20070307/libavcodec/amr_float 目录。

（8）下载 amr wb float，同样应该从地址 http://www.3gpp.org/ftp/Specs/archive/26_series/26.204/中选择最新版本下载。

（9）将压缩文件解压，将 c-code 目录中的所有文件复制到 ffmpeg-0.cvs20070307/libavcodec/amrwb_float 目录。

（10）进入到 FFmpeg 源码目录，使用命令“make”编译。使用命令如下：

```
cd  ～/ffmpeg-0.cvs20070307 #改变当前目录为 FFmpeg 源码目录
make                        #开始编译源码
```

（11）使用“sudo checkinstall -D make install”安装，这样成功编译安装了增强版的 FFmpeg。如果编译中出现如下错误：

```
install:'cannot stat 'doc/ffmpeg.1' No such file or directory
.. ffplay.1 ..
.. ffserver.1..
make: *** Error 1
**** Installation failed. Aborting package creation
Cleaning up...OK
Bye.
```

可以运行如下命令来修复这个错误，然后重新编译即可。

```
sudo apt-get install libx264-dev checkinstall
```

13.3.4 使用 FFmpeg 转换视频文件

虽然使用前面介绍的软件可以完成视频的转换，但是这些软件提供的是模式化的转换。对于高级用户或者有特殊需要的用户而言，可能更喜欢精确控制转换的细节。这样的要求，普通的转换软件就无能为力了。为了完成这样的任务，就需要使用命令行的方式。因为命令行程序

一般均提供了复杂繁多的参数供用户选择。Linux 中可以完成此项操作的程序有 Mencoder 和 FFmpeg。下面就简单地介绍一下如何使用这两条命令精确控制视频转换效果。

1．将 avi 转换为 mp4

一般而言，网上下载得到的许多视频文件均是 avi 格式。这种格式的视频普通的 mp4 是无法播放的，所以需要先将其转换为 mp4 格式。Avidemux 提供了模式化的选择。但也可以用如下的命令精确控制转换的细节。

```
    ffmpeg -i "Wall-E.avi" -vcodec xvid -s 320x240 -r 25 -b 800 -acodec aac -ac
1 -ar 48000 -ab 128 -vol 200 -f mp4 "～/movie/Wall-E.mp4"
```

参数解释如下：

- -i " Wall-E.avi "：源文件名，指定要转换的电影的目录和文件名。
- -vcodec xvid：转换文件使用 XVID 编码，不能更改。
- -s 320x240：转换后文件的分辨率，应该依据源文件的比例进行修改。
- -r 25：每秒帧数。
- -b 800：视频码率，数字越大，质量越高。还可以用动态码率如“-qscale 4”（这个数字越小，视频质量越高）。
- -acodec aac：音频编码指定使用 AAC
- -ac 1：转换后文件的声道数。可以为 1 或 2。
- -ar 48000：声音的采样频率。
- -ab 128：音频码率，一般选择 32、64、96、128。
- -vol 200：转换后音频音量提高 200%。
- -f mp4：转换为 mp4 格式文件。
- "tt.mp4"：转换目标文件名。

意思是将 Wall-E.avi 文件转换为 Wall-E.mp4 文件。转换为视频大小为 320x240、每秒 25 帧、视频码率为 800、音频编码使用 AAC、音频码率为 128、音量提高 200%。

2．将 rmvb 转换为 avi

现在，rmvb 格式的视频格式很常见，因为其提供了清晰的画质与超高的压缩比。但很多转换软件是无法直接将 rmvb 格式转换为 mp4 格式的，所以可能需要先将 rmvb 格式转换为 avi。实现此转换使用如下命令即可。

```
    mencoder CSI.rmvb -ovc lavc -oac mp3lame -o CSI.avi
```

- CSI.rmvb：欲转换的源文件。
- -ovc lavc：转换目标视频编码。
- -oac mp3lame：转换目标音频编码格式。
- -o CSI.avi：转换后的文件名。

3．合并两个视频

一般情况，下载的电影会分为两个部分。但是有时为了方便起见，需要将这两个部分合并为一个文件，就需要使用如下命令。

```
    mencoder CSI1.avi CSI2.avi -ovc copy -oac copy -o all.avi
```

- CSI1.avi CSI2.avi：要合并的源文件，可以有多个文件。合并后按输入时的顺序依次合并。
- -ovc copy：目标视频编码格式复制了源文件的格式。
- -oac copy：目标音频编码格式复制了源文件的格式。
- -o CSI.avi：合并后的文件名。

意思是保持电影原始码率，并将两个部分依次连接为一个文件。

13.4 小结

本章介绍了 Linux 平台多媒体软件的使用，主要包括 Linux 中著名的视频播放器 Mplayer，音频播放器 Audacious，以及功能强大的视频处理软件 Avidemux 的使用，也为高级用户简单介绍了使用命令行转换视频文件的方法。本章中比较高阶的内容包括编译 FFmpeg 及使用 FFmpeg 转换视频的方法，及 Avidemux 的使用，这些内容多加实践才会有比较深入的掌握。Mplayer 及 Audacious 的使用对于初级用户来说则是比较实用，也是比较比较容易掌握的。

第 14 章　图形图像软件

随着数码相机的逐渐普及，在电脑中管理数码照片成了普通用户经常性的操作。普通用户可能已经习惯了 Windows 中的图形图像软件的使用模式。不过，一般来说，Windows 中的软件，在 Linux 中均有相对应的免费软件可实现其功能。本章将介绍这些优秀的 Linux 平台的图形图像软件。经过本章的阅读，读者可以具备图片管理、图片处理及矢量绘图的相关知识。

14.1　图片浏览软件 gThumb

图片浏览软件是拥有数码相机的用户最经常使用的软件。有没有一款使用方便、功能强大的图片浏览软件，直接关系到用户体验。不过，既然来到了 Linux，就先忘掉 ACD See，来看看 gThumb，其功能同样强大。

14.1.1　软件简介

Ubuntu 默认打开图像的软件是 Eye of GNOME（图像查看器），但这款软件有些缺点，如只能浏览图片，无法对图片做简单的处理，这就限制了其使用范围。在这种情况下，寻找一款更好的，有简单处理功能的图片浏览器就是当务之急。我们首先推荐的是 gThumb。这款软件不只可以浏览图片，还可以对图片进行简单的处理，如更改图片大小、裁剪图片、更改其对比度、更改色度、消除红眼等功能。

1．软件安装

由于 Ubuntu 默认并未安装该软件，但软件仓库中有该软件，所以可使用“sudo apt-get install gThumb”进行安装。安装成功后，即可在终端中输入 gThumb 或选择 “应用程序”|“图像”|“gThumb 图像浏览器”命令，打开该软件。gThumb 的主界面如图 14.1 所示。



图 14.1　gThumb 主界面

2．将图片文件默认打开方式更改为 gThumb

安装后，系统在每次打开图片文件时仍然使用原来系统默认的图片浏览器（EOG）打开图像。所以要做的第一步工作就是改变系统的默认行为，让系统默认使用 gTumb 来打开图片文件。其设置步骤如下。

（1）单击系统主菜单“位置”中的任意项，如主目录，打开文件浏览器。

（2）找到任一图片文件的目录，如/usr/share/backgrounds 目录。右击任一图片，在弹出的菜单中选择“Properties（属性）”菜单项，弹出对话框，如图 14.2 所示。

（3）在弹出的对话框中单击 Open With 标签，打开文件打开方式设置标签页。

（4）该标签页的列表框中有许多软件，这些软件是系统所有可打开该图片的软件。在列表中选择 gThumb Image Viewer 选项。

（5）单击 Close（关闭）按钮确认。

现在系统就会默认在 gThumb 中打开 JPG 图片了。在文件浏览器中双击该图片，应该可以看到该图片在 gThumb 中被打开。

还可参照上述方法更改系统默认打开其他格式图片文件的方式，如 PNG、BMP、GIF 等。修改后，所有的图片文件就会默认采用 gThumb 软件打开了。打开图片文件后，就可以使用 gThumb 的强大功能。下面介绍一些 gThumb 的功能。



图 14.2　设置图片的默认打开方式

14.1.2　图片浏览

gThumb 主要功能是用于图片浏览。所以，首先介绍其图片浏览功能。gThumb 图片浏览有两种方式，一是普通的图片浏览，二是幻灯方式图片浏览。下面分别予以介绍。

1．浏览图片

单击任何图片，则在 gThumb 中打开图片，如图 14.1 所示。在浏览过程中，可以使用“空格”键切换到下一张，按下 Backspace 键切换到上一张。切换图片的操作也可以单击工具栏中相关按钮来实现。

单击工具栏“放大”、“缩小”按钮可以放大和缩小图片的显示效果。缩放范围从 5%到 10000%，也就是说可以缩小图片到原来的 1/20 或者放大图片 100 倍显示。但是这种方式只是在显示效果上缩放图片，实际图片大小并没有任何改变。

2．幻灯方式浏览

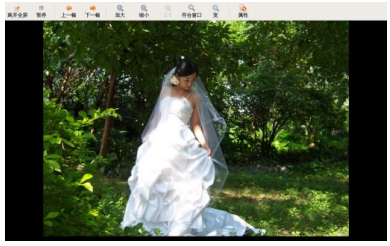


图 14.3　幻灯方式浏览

如果图片较多，而且自己不想动手按鼠标或键盘，希望让图片自动显示，就可以使用该功能。选择 gThumb 菜单的“查看”|“幻灯片播放”命令，则图片开始以幻灯方式全屏自动播放，如图 14.3 所示。

默认方式是每 4 秒钟切换一个图片。如果觉得图片切换太慢，可以使用“空格”键手动切换到下一张图片，按下 Backspace 键切换到上一张图片。图片显示结束后，gThump 会自动退出幻灯模式，回到正常浏览界面。

14.1.3　图片处理

我们推荐 gThumb 的理由是其功能强大。本小节介绍的内容是使用 gThumb 进行简单的图片处理工作，包括将图片设置为壁纸、更改图片格式，以及更改图片大小等。

1．将图片设置为桌面壁纸

在图片浏览过程中，如果发现比较漂亮的图片，可能许多人都想用其来作为桌面壁纸。这在 gThumb 中是很容易的。只需选择“工具”|“将图片设置为桌面壁纸”命令，选择“居中”或“平铺”命令，这时就可以看到当前正在浏览的图片成为桌面背景。

2．更改图片格式

gThumb 提供了四种图片格式之间的转换，分别是 JPEG、PNG、TGA、TIFF。也就是说可

以使用 gThumb 将图片转换为这四种格式中的任意一种。其具体设置步骤如下：

（1）单击“工具”菜单，选择“转换格式”命令，出现如图 14.4 所示对话框；

（2）如果要将图片转换为 PNG 格式，在“图像类型”单选框中选择 PNG 单选按钮，单击“转换”按钮完成。即将当前浏览的图片转换为 PNG 格式。该项功能可以将图片转换为 TIFF、TGA、JPEG、PNG 等格式。

3．改变图片的大小

如果图片尺寸比较大，则其打开速度会比较慢，通过网络发送也比较耗费时间。所以，在很多时候，需要将图片的尺寸缩小以方便使用。这在 gThumb 中也是很容易实现的功能，其具体设置步骤如下。

（1）单击菜单“图像”|“改变大小”命令，出现如图 14.5 所示的对话框。

（2）在“新的大小”下面的微调框中输入要将图片改变的目标大小。这个大小可以是像素值，如宽度设置为“640”，高度设置为“480”，则将当前打开的图片大小设置为 640×480 像素。



也可以将“高度”后面的下拉框设置为“%”，意思是将图片大小按照百分比缩放。这样就可以改变图片的大小为原来的百分之多少，如输入 60 则改变图片为原图的 60%。

（3）单击“缩放图像”按钮，图片就会按照设置的大小改变。要记得在 gThumb 中单击“保存”工具栏按钮，将更改后的结果保存。



图 14.4　图片格式转换



图 14.5　改变图片大小

4．裁剪图片

有时候，数码照片在拍摄时会摄入一些无用的内容；或者有时要改变图片的布局，这时就需要对图片进行裁剪操作，将不需要的部分剔除。在 gThumb 中，具体设置步骤如下。

（1）选择“图像”|“裁剪”命令，出现图 14.5 对话框。

（2）用鼠标勾选图片中需要保留的部分，就是图中被虚线框住的部分。虚线框外的内容将会被裁剪。

（3）单击“裁剪”按钮，图片被按虚线裁剪，如图 14.6 所示。

（4）单击“应用”按钮，结束裁剪过程。

（5）同样要在 gThumb 中单击工具栏“保存”按钮，将更改后的结果保存。

5．旋转图片

在拍摄照片过程中，为了拍摄想要的内容，有时不得不将相机进行旋转。这样拍摄的照片

在电脑中浏览时会出现方向错误，如人物的脚朝上、头朝下的情况。这时，需要进行图片旋转工作，具体方法是选择“图像”|“转换”命令，在打开的对话框中依照实际情况选择“向右旋转”或“向左旋转”等项，图片会按照选定操作进行更改。

6．更改颜色平衡

由于相机的白平衡功能还不尽完美，所以拍摄的照片难免有偏色现象出现。不过，在 gThumb 中调整偏色也是比较容易的。假设本次要将照片设置为一种怀旧色，其具体步骤如下。

（1）选择“图像”|“色彩平衡”命令，出现如图 14.7 所示对话框。

（2）将“蓝绿色－红色”滑块向右拉动，使之变大。

（3）将“紫红色－绿色”和“黄色－蓝色”后面的滑块向左拉动，使之变小。

在调整颜色的过程中，可以随时单击“预览”按钮，使更改的效果显示在主界面中，以便查看。达到满意的效果后，单击“确定”按钮确认退出。



图 14.6　裁剪图片



图 14.7　设置颜色平衡

7．其他功能

gThumb 的图像菜单中还有许多菜单项，可以实现多种功能。如“去除彩度”菜单项是将图片变为黑白、“底片效果”原理是将图片反色，效果是将图片变换为类似胶片底板的颜色、调整对比度等。这些功能操作步骤与前述其他功能基本相同，打开相应的功能设置对话框后按照提示进行操作即可，这里就不再进行介绍了。

14.1.4　制作网络相册

gThumb 还有几项比较强大的功能，其中一项是将图片制作成网络相册。所谓的网络相册，就是将多张缩略图显示在一个网页上，单击任一缩略图可打开相对应的原始图片。其效果如图 14.8 所示。

在制作网络相册之前，先介绍一下 gThumb 的两种模式，一是图片浏览模式，一是图片管理模式。如图 14.1 所示主界面中被称为浏览模式，在该模式中主要完成图片浏览、图片处理功能。而图 14.9 所示称为图片管理模式，主要完成图片组织管理等功能。

单击图 14.1 所示工具栏“关闭”按钮，即关闭图片浏览模式，打开图片管理模式。制作网络相册需要在图片管理模式中进行。

确保处于图片管理模式，然后参照如下步骤，就可将相应的图片制作为网络相册。

（1）将图 14.9 右侧列表框中想要加入网络相册的图片全部选中。在状态栏会显示相关的信息，如“18 幅图像，选中了 13 幅”等。

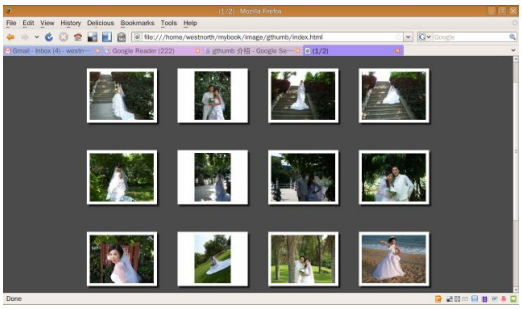


图 14.8　网络相册效果

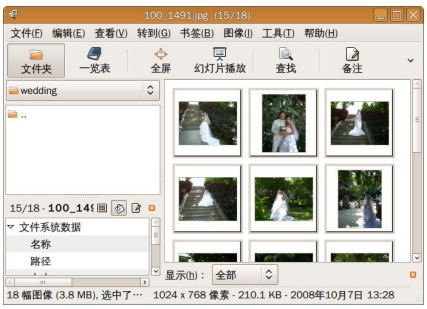


图 14.9　gThumb 图片管理窗口

（2）选择“工具”|“创建网络相册”命令，会弹出一个设置对话框，在其中可以设置该网络相册保存的位置等项。

（3）该对话框下方的“主题”文本框在整个设置中是极其重要的一项，一定要单击其后的“…”按钮，选择一款合适的主题，才能进行创建网络相册的工作。

（4）设置完成后，单击“保存”按钮就可让 gThumb 创建网络相册。

创建完成后，进入保存网络相册的目录，打开创建的 HTML 文件，即可查看网络相册。效果应该如图 14.8 所示。

也可以先选中图片，然后选择“工具”|“创建索引图像”命令制作索引图像。索引图像是将选中的图片全显示在网页上，图片下有该图片的文件名，单击该图片可打开原始图片文件。效果如图 14.10 所示。

14.2　系统默认的图片浏览软件 Eye of GNOME

Eye of GNOME，简称 eog，是 Ubuntu 系统默认的图片浏览软件。这款软件的缺点是功能有点过于简单，而且加载图片的速度比较慢。

打开 eog 的方法很简单，选择“应用程序”|“图像”|“图像浏览”命令即可。如果在命令行中操作，则输入 eog 即可打开。其界面如图 14.11 所示。

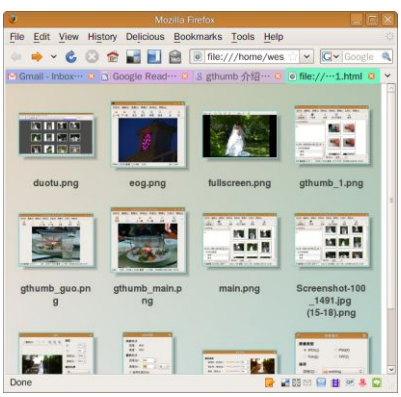


图 14.10　图片索引效果

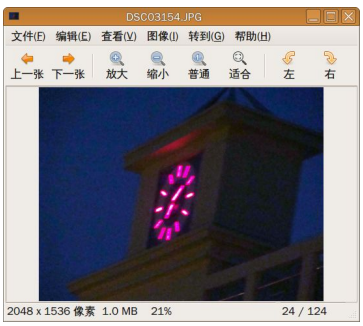


图 14.11　Eye of GNOME 的界面

作为默认的图片浏览程序，eog 也有一项优点，就是简洁。其所有功能都列在工具栏中，

如放大、缩小图片等功能。而且 eog 将旋转图片的功能也列于工具栏，所以对于经常需要旋转图片的读者可能比较适用。

用键盘切换图片的方式与 gThumb 是一样的，按下空格键是显示下一张图片，而按下 Backspace 键则显示上一张图片，按下 F5 键是幻灯方式播放当前目录的图片。

14.3 图片管理软件 Picasa

图片管理功能是指以多种模式组织图片，并以某种使用户方便查看的方式排列图片的功能。同时图片管理还包括从相机或其他存储设备中导入图片，以及将图片上传至网络等功能。具备这些能力的软件在 Linux 中有许多，但其中功能最为强大的软件应该算是 Google 公司出品的软件 Picase 了。

14.3.1 功能简介

Picase 是一款功能极其强大的图片管理软件，多次被评为最佳图片管理软件。其由 Google 出品，主要特色也与 Google 一样，速度快，包括图片显示速度及图片搜索速度均很快速。

Picasa 软件安装后，首先会在系统中“我的文档”（在 Linux 中是～/Documents）与“图片”（～/Pictures）目录中扫描所有图片，并将这些图片分门别类列出来。如果这两个文件夹中增加了图片文件，也会很快在软件中反映出来。Picase 打开图片的速度极快，而且内置了强大的搜索功能，搜索图片速度极快。除了速度快外，Picasa 还具有如下特色。

- 与网络相册的整合，使得图片上上传功能变得很简单。只要拥有一个免费的 Google 账号，就可以方便地将图片上传到 Google 相册中，与网上的朋友们共享。还可以轻易地实现与网络相册的同步工作。
- 图片拼贴画工具，可以制作出图片重叠、图片方格等功能。制作出的图片效果如图 14.12 所示。
- 方便地给图片添加文字及水印功能。



图 14.12 图片拼贴画效果

14.3.2 安装

Picase 的安装有两种方法，另一是经过设置后直接用 apt-get 安装，另一是直接下载 Deb 包文件到本机安装。相对来说，可能使用 apt-get 安装的软件版本低些，而自己下载的 Deb 包可能会是最新版本。读者可以根据自己的需要灵活选择。下面分别介绍如何使用这两种方法进行安装。

1．使用 apt-get 进行安装

使用 apt-get 安装的原理是添加一个有 Picasa 软件的源服务器，然后就可直接使用 apt-get 进行安装了。具体步骤如下。

（1）首先应该添加一个拥有 Picasa 软件的源服务器。本例采用 Google 的官方服务器。方法是用命令 sudo gedit /etc/apt/source.list 打开源列表，在其中加入 deb http://dl.google.com/linux/deb/ testing non-free。

（2）使用命令 wget https://dl-ssl.google.com/linux/linux_signing_key.pub 下载密钥。将下载的

密钥文件放于主目录中。

（3）将密钥加入 apt 中，使用命令 sudo apt-key add linux_signing_key.pub。

（4）使用命令 sudo apt-get update;sudo apt-get install picasa 即可。这条命令的意思是刷新源列表，然后进行安装。

2．直接下载 Deb 包

还有一种方法是直接下载 Deb 文件安装，在 http://Picasa.google.com/linux/下载 linux 版的 Picasa 软件的 Deb 包，然后双击下载的文件即可进行安装过程。

14.3.3　设置

Picasa 在使用前有两项内容需要设置，一是字体，是为了让 Picasa 显示得更为美观些；另一是扫描文件夹，是为了控制 Picasa 管理的目录，也就是说让 Picasa 显示指定目录中的图片。

1．设置字体

到目前为止，Picasa 的所谓 Linux 版其实仍然是通过 Wine 运行的 Windows 版软件。所以，其在许多方面还会有些让人不太满意的地方。首当其冲的问题就是字体显示。所以应首先解决字体显示的问题。不过，Google 也应该认识到这个问题，所以给出了字体设置程序，这也使得字体设置容易了不少。具体的设置步骤如下。

（1）选择“应用程序”|“图像”|“Picasa”|“Picasa 字体设置”命令，出现如图 14.13 所示的对话框。

（2）在图中单击右侧 Add 按钮，将 WenQuanYi ZenHei、Dejavu Serif 等字体加入其中 Linded Fonts 列表中，Picasa 就可使用这些字体显示。

2．设置扫描目录

在进入 Picasa 后，程序会自动扫描一些目录，将其中的图片纳入 Picasa 管理。程序默认只扫描两个文件夹，用户主目录中的 Documents 和 Pictures 文件夹。但是如果用户有大量图片保存在其他目录，就需要手动将这些目录也纳入 Picasa 管理。

假设要加入黑色眼睛关注的目录是 My Documents/Photos（这是由于该软件是从 Windows 移植过来的，还带有 Windows 的烙印，在 Linux 中实际的目录是～/Documents/Photos）。将该目录加入 Picasa 中的具体步骤如下。

（1）选择 Picasa 菜单的 File|Add file to Picasa 命令，出现如图 14.14 所示的对话框。

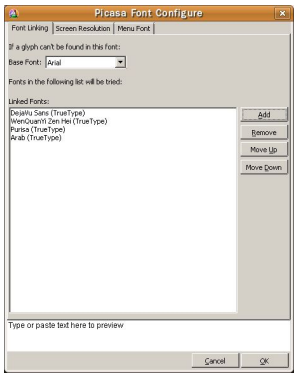


图 14.13　设置字体

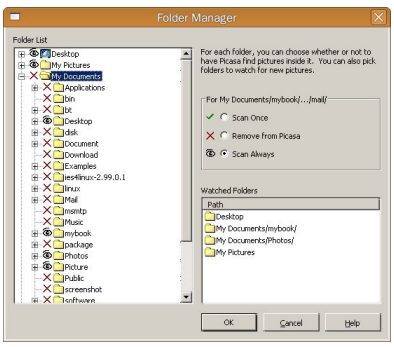


图 14.14　扫描目录设置

说 明

图左侧是一个目录树，列出了当前系统中所有目录。在目录前有各种不同的符号，代表不同的意思。绿色的对号代表 Picasa 会在第一次启动时扫描一次的目录；红色的小叉代表不扫描该目录中的文件；黑色的眼睛意思是 Picasa 将保持对该目录的关注。如果该目录的文件发生变动，Picasa 将第一时间反映其变动。

（2）在左边的目录列表中找到 My Documents/Photos 选项，单击选中该目录。

（3）选择右面 Scan always 单选按钮，可以看到 My Document/Photos 已经出现在右下方的 Watched Folders 列表框中。可以按前述步骤将所有要 Picasa 关注的目录均加入到 Watched Folders 列表框。

（4）单击 OK 按钮完成设置。这样，如果这些目录中文件发生了变动，如增加了图片，则 Picasa 会很快注意到，然后将新增加的图片显示出来。

14.3.4 使用简介

Picasa 功能比较强大，所以其界面也比较复杂。其界面也可分为三种模式：一个模式是图片管理模式（也称为主模式）；另一个模式是时间线浏览模式，该模式界面很漂亮，可以按照时间线浏览图片；第三个模式是图片浏览模式，可以在其中进行图片浏览与处理工作。下面分别予以介绍。

1. 图片管理模式

Picasa 启动后会首先进入图片管理界面（主界面），如图 14.15 所示。图中左侧列表中显示了包含图片的目录，目录边的数字说明该目录中包含图片文件的数目。右侧显示了其中一个目录中的图片文件。在右侧图片上方有一排按钮，单击有文字 Play 的按钮后会以幻灯方式放映当前目录中的图片。第二个按钮的功能是创建图片拼贴画。

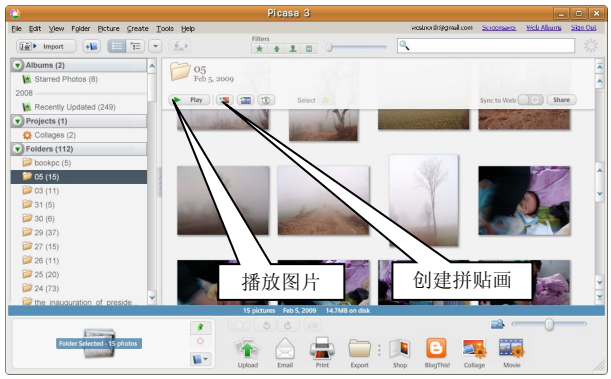


图 14.15　Picasa 主界面

在界面的最底端，有一排按钮。可以执行文件上传、电子邮件发送、文件打印、导出等功能。这些功能将在后面相关内容进行详细介绍。

2. 时间线浏览模式

在 Picasa 任意界面按下 Ctrl+5 组合键，则开启时间线浏览方式。这种方式提供了美观的界面，可供用户按时间顺序管理图片，界面如图 14.16 所示。可以用鼠标拖动图片使其沿着圆弧形的轨迹移动。每张图片代表一个目录，双击图片可用幻灯方式浏览该目录中的图片。

3．图片浏览模式

双击图 14.17 右侧的任一张图片，进入图片浏览模式，界面如图 14.16 所示。左侧的 8 个按钮其功能分别是 Crop（图片剪切功能）、Straighten（稍微旋转图片）、Redeye（消除红眼）、I’m feeling lucky（自动调整图片）、Auto Contrast（自动调整对比度）、Auto Color（自动调整色彩）、Retouch（图章工具）、Text（在图片中加入文字）。



图 14.16　时间线浏览模式

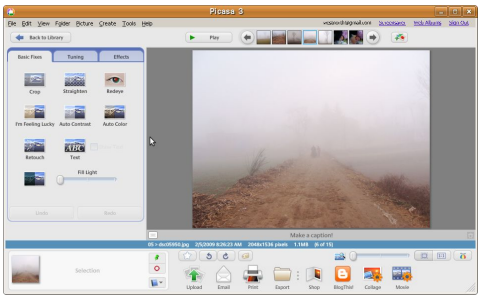


图 14.17　图片浏览界面

下面以 Text（给图片添加文字）功能为例，介绍其用法。其具体步骤如下。

（1）单击如图 14.17 所示的 Text 按钮，出现图 14.18 所示的文字编辑界面。

（2）用鼠标在图片中单击，出现一个灰色的圆角矩形，在其中输入文字即可。

（3）图 14.18 界面左侧还有调整文字属性的设置项，可以调整字体、大小、颜色等项。如图 14.18 标注所示。

（4）将鼠标移动到文字上，会出现一个灰色的圆圈。移动灰色圆圈即可移动文字位置，也可放置圆圈以旋转文字。

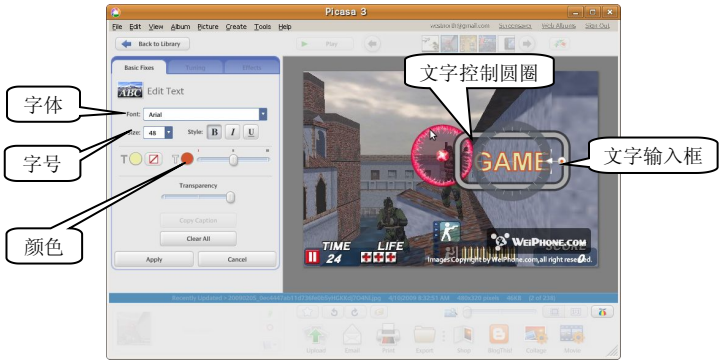


图 14.18　给图片添加文字

4．色度及色温调节

如果单击图 14.17 所示对话框左侧 Tuning 标签，则打开色度调整页。该页提供了色度以及色温调节功能，界面如图 14.19 所示。

5．特效设置

如果选择图 14.17 所示对话框左侧 Effects 标签，则打开特效设置页。该页在左侧列出了 9 张图片，分别代表九种特效所导致的结果，觉得哪一个比较满意，单击该图片即可。这些特效包括图片锐化、使图片成为怀旧色调、将图片变为黑白、使图片色彩变暖、电影胶片效果等。而且

这些项目的设置均比较简单，就不再详细介绍了。如图 14.20 所示是边缘模糊特效设置后的效果。

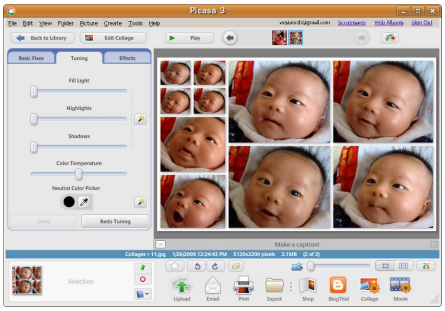


图 14.19　色度调节标签页



图 14.20　特效设置页

14.3.5　网络相关功能

在 Picasa 右上角有一个 Sign to Web Albums 超链接，单击其输入 Google 的用户名和密码可登录 Google 网络。登录成功后就可使用其与执行相关的网络操作，如同步网络相册等功能。Picasa 可以执行的网络操作如下所示。

1．本地电脑与网络相册进行图片同步

在 Picasa 程序界面的下方，有一个“Upload”按钮。如果使用 Google 账号登录了网络，单击其可将当前选中的图片上传到网络相册中。也可以选择 File|Import From Web Album 命令来下载网络相册中的图片。具体上传的步骤如下所示。

（1）首先回到图 14.15 所示的 Picasa 主界面。如果在其他界面，只需单击 Back to Library 按钮即可回到该界面。

（2）选中要上传到网络的照片。

（3）单击 Upload 按钮后出现如图 14.20 所示图片上传对话框。

（4）Picasa 会创建以图片所在目录为名的影集。如图 14.21 所示，目录名为 05，则会将该图片上传到网络相册的 05 目录中。也可单击 New 按钮新建网络目录。



图 14.21　图片上传

（5）单击 Size to Upload 下拉列表框可以设置上传图片的尺寸。可以保留原始尺寸，也可将图片尺寸缩小以节省网络空间。

（6）单击 Public Album 单选框则将图片浏览权限设置为公开，意思是所有人均可浏览。而

单击 Unlisted Album 按钮则不允许未经授权的用户浏览当前上传的图片。

（7）设置结束后，单击 Upload 按钮上传图片。上传成功，则会显示如图 14.22 所示对话框。

2．将图片通过电子邮件发送

许多人在拍摄了漂亮的图片后，都喜欢将这些图片发送给朋友共享。在 Picasa 中，将图片以电子邮件方式发送给朋友是很容易的。具体步骤如下。

（1）首先选中要发送的图片。可以是一张，也可以是多张。

（2）单击界面下方的 Email 图标，出现如图 14.23 所示图片。选择其中的 Google Mail 选项即可。

（3）现在出现如图 14.24 所示界面。在 To 文本框中输入收件人地址，然后单击“Send”按钮就可以将选中的图片以附件形式发送给收件人。



图 14.22　图片上传成功

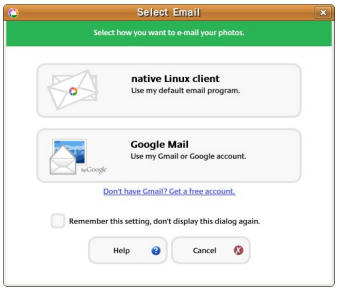


图 14.23　选择邮件发送方式



图 14.24　通过邮件发送图片

3．将图片发到博客

现在，随着博客的普及，许多朋友都拥有属于自己的博客。但俗话说“一张合适的图片效果好于 500 个文字”，所以在博客中使用图片也是很普遍的事。许多时候在博客中上传图片并不是很方便的。但现在使用 Picasa 就方便多了，可以直接将本地图片贴在博客中。具体步骤如下所示。

（1）选中要发布在 blog 中的图片。

（2）在 Picasa 界面下方找到 图标，单击该图标，出现设置图片位置设置对话框（前提是已经用 Google 账号登录），如图 14.25 所示。主要是选择图片在网页中出现的位置，在其中选择适当的位置即可。然后单击 Continu 按钮，出现如图 14.26 所示对话框。

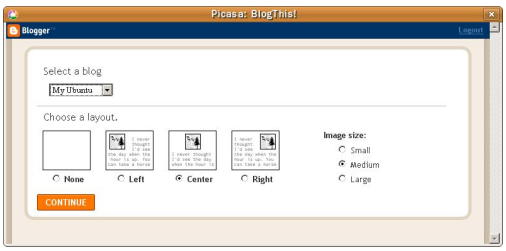


图 14.25　设置图片的位置



图 14.26　将图片发布到博客

（3）将要写的文字写在正文区，再在 Title 文本框中输入博客的标题，单击 PUBLISH POST 按钮即可将图片及文字发布到博客中。



使用这项功能只能将图片发送到 Google 的 blog 中，也就是说，博客地址应该类似于 zhangsan.blogspot.com 这样的才可以通过 Picasa 发布。

14.3.6 其他功能

Picasa 还有一项很有特色的功能是制作拼贴画。这项功能可以将多幅照片组合到一张图片中，以达到特定的效果。实际操作的结果证明，这种方式制作的图片有很强的实用性。作为一款图像管理软件，导入图片的功能也是必不可少的。本节介绍的就是如何创建图片拼贴画及如何从其他设备中导入图片。

1. 创建图片拼贴画

所谓的图片拼贴画，就是将多幅图片组合到一张图片中的过程。这样，图片会有一种很特别的感觉，在广告宣传中比较常用。在 Picasa 中设置也比较简单，下面是具体的步骤。

（1）选中要创建拼贴画的图片，这也是制作任何效果之前要做的事。不过，制作拼贴画必须选择多张图片。

（2）单击界面下边的 collage 选项，开始制作拼贴画。界面如图 14.27 所示，左侧标示的下拉框可以设置组成拼贴画的模式，右边是使用该方式生成的图片预览。Picasa 提供了许多风格的拼贴画模式可供选择。一些模式的简单介绍如下。

- White border 模式是给图片加上一个白色的边框。
- Draw shadows 模式是给图片加上一个阴影效果，以增强立体感。
- 在选用 Picture Pile 模式时，还可以单击图片移动图片的位置，以使得图片摆设地更加美观。
- 在选用 Mosaic 模式时，每张图片之间的距离也可以改变，以达到不同的效果。效果如图 14.28 所示。



图 14.27 创建拼贴画对话框



图 14.28 Mosaic 效果

- 图片还可以使用拍立得相机模式，如在 Contact Sheet 中将照片改为拍立得照片模式，如图 14.29 所示。

2. 导入图片

随着数码相机的逐渐普及，从相机中导入图片也成为日常操作之一。Picasa 同样提供了强

大的图片导入功能，具体导入步骤如下。

（1）回到图 14.15 所示主界面，单击有相机图案的 Important 按钮，或选择 File|Import from 命令打开导入数据页面，显示如图 14.30 所示对话框。

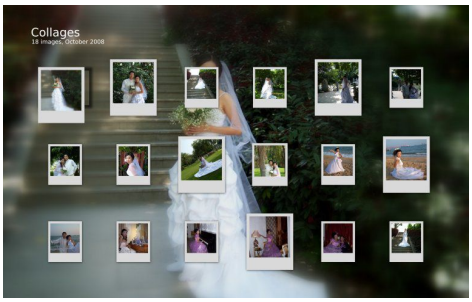


图 14.29　Contact Sheet 效果

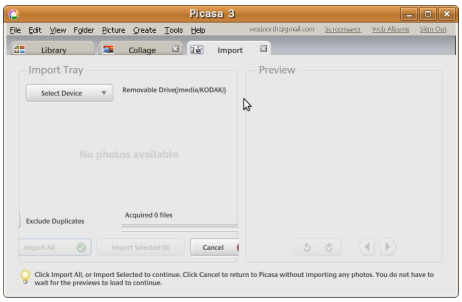


图 14.30　选择导入设备

（2）在该页面左上方 Select Device 下拉列表框选择适当的设备，系统会从该设备中读取图片的资料。等待一会，读取结束后，会将所有检测到图片列出来。

（3）如果 Picasa 读取的图片就是要导入的图片，单击页面下方的 Import all 按钮，开始图片导入过程，出现如图 14.31 所示的对话框。

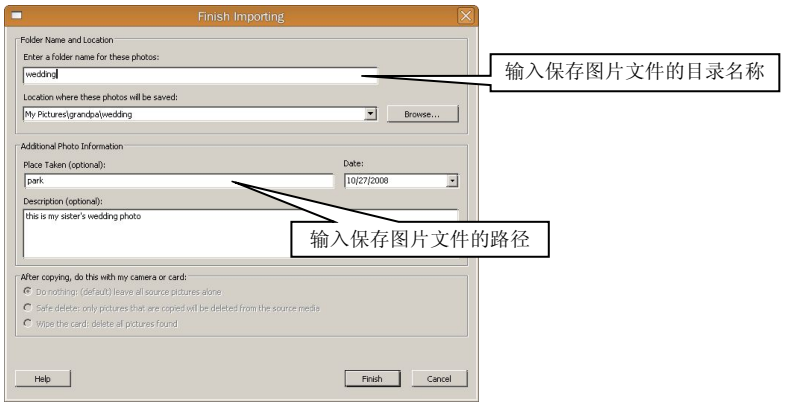


图 14.31　导入图片

（4）在图 14.31 对话框中 Enter a folder name for these photoes 文本框中输入保存这些图片的目录名，如输入 wedding。

（5）在 Location where these photos will be saved 文本框中输入要保存这些图片的路径，如输入 My Pictures\grandpa\wedding。

（6）单击 Finish 按钮结束导入，则将导入的图片就保存在～/Picture/wedding/wedding 目录中。

14.4　系统默认的图片管理器 F-Spot

F-Spot 是 Ubunu 默认的图片管理软件。F-Spot 也是 Linux 中一款功能比较强大的图片管理软件，其可在导入图片的过程中，自动按照图片拍摄的日期进行分类。在显示图片过程中，还能自动根据重力方向，自动调整图片的方向。

F-Spot 的主要特色是提供了类似智能标记的分类，可以方便地对图片文件进行分类管理。而且其图片浏览采用预加载功能，所以感觉速度还是令人满意的。在网络应用方面，F_Spot 可以方便地将图片上传至 Flickr、Zooomr 等图片网站，这个功能还是比较实用的。

1．界面

如果将带有图片的移动磁盘或 U 盘、相机之类连接到电脑，系统弹出的提示如图 14.32 所示。单击 Open F-Spot Photo Manager 按钮即可打开 F-Spot 软件，其界面类似于如图 14.33 所示。



图 14.32　插入 U 盘后的提示



图 14.33　F-Spot 界面

界面左侧有“收藏”、“隐藏”、“地点”等几个标记。这几个标记相当于图片的分类，用户可以将图片放入该标记中，单击该标记即可查看该分类的所有图片。界面右侧是图片预览。

2．标记管理

在 F-Spot 中图片不是按照其保存的目录来分类、而是按标记来管理的。所以，要有效管理图片，就应该先给图片加上标记。添加标记的具体步骤如下。

（1）在右侧图片列表中右击图片，在弹出的菜单中选择“附加标记”命令。

（2）该子菜单后出现级联菜单，内容是界面左侧的各个标记，选择任一标记，则可将图片加入该标记中。

如果要删除图片的标记，右击该图片，选择“删除标记”命令即可。

所有的系统标记均可进行管理，如添加新标记种类、删除原有标记、在某标记下添加子标记功能等。方法也很简单，直接右击图 14.33 左侧列表框中的任一标记，在弹出的菜单中选择相关选项即可。

3．图片编辑

在图 14.33 所示界面中，单击“编辑图像”工具栏按钮或双击任一图片即可开始编辑图像，其界面如图 14.34 所示。



图 14.34　F－Spot 图片编辑界面

该界面中与主界面的主要区别是界面右下方多了一排图片编辑按钮。这些按钮可以实现图片剪切、调整颜色、消除红眼、将照片做旧、使边缘模糊等效果。

4．上传图片

前面提到过，F-Spot 有项很实用的功能就是将图片上传到多个图片网站。下面就来介绍上传图片的步骤。

（1）选中图片。这是任何操作进行前必须要做的一项准备工作。

（2）选择 F-Spot 菜单的“照片”|“导出到”|“Flickr”命令，出现如图 14.35 所示的对话

框。可以看到，选中的图片已经出现在对话框左侧列表框中。

（3）单击“身份验证”按钮。这是因为如果使用程序向 Flickr 网站上传图片，Flickr 会要求用户验证。

（4）可设置相关的选项，如“权限”等。设置完成后，单击“Export”按钮上传图片。



图 14.35　上传图片

14.5　图像处理软件 GIMP

提起图像处理软件，许多人马上会想起 PhotoShop，但其高昂的价格将许多潜在的使用者挡在了门外。不过，本着自由、共享的精神，Linux 平台中也有一款同样强大且免费使用的软件 GIMP。GIMP 提供了与 PhotoShop 类似的功能，可以轻松实现许多复杂专业的效果。本节将介绍 GIMP 的相关特点及其简单的使用。

14.5.1　简介

GIMP 的全称是 GNU Image Manipulation Program（GNU 图像处理程序）。1995 年，美国加州大学柏克利分校的学生 Spencer Kimball 和 Peter Matt 开始了他们的学期项目，这就是 GIMP 的最初版本。1996 年，GIMP 发布了 0.54 版，该版已经具备了初步的功能，如可以使用多种方式选取图片，支持 JPEG、GIF、PNG 等图片格式，有文字输入及简单的滤镜，而且 Linux 的吉祥物 Tux 就是用这个版本的 GIMP 做出来的。



图 14.36　GIMP 的吉祥物

该项目在 1997 年被命名为 GNU Image Manipulation Program。在他们毕业后，该项目正式成为 GNU 项目中的一部分。有许多的开发者参与了 GIMP 的开发中，现在其稳定版已经发布到 2.6 版。其吉祥物是只可爱的小狐狸，如图 14.36 所示。

GIMP 是一款功能强大的图片处理软件，功能不逊色于专业的图片处理王者 PhotoShop。GIMP 支持图层、通道等高级特性，除了自身附带的滤镜外，通过 Wine 还可以使用 Photoshop 的滤镜。其主要特点如下。

- 支持图层和通道。可以读取 PhotoShop 的 PSD 文件，并正确识别图层、通道等信息。
- 对 Alpha 通道提供了强大的支持。支持多种图片文件格式：如 GIF、JPEG、PNG、XPM、TIFF、TGA、MPEG、PS、PDF、PCX、BMP 等。可读取、显示、转换、保存多种文件格式。
- 丰富的图像处理工具。GIMP 中提供画笔、铅笔、喷枪、图章等强大的工具，可以轻松地在图片中添加线条。
- 先进的脚本化处理能力。可以编写自己的脚本来自动实现某些复杂的功能。
- 多级撤销功能。撤销能力只与内存设置有关，设置的内存足够大的情况下，甚至可以无限指定撤销次数。
- 灵活强大的变换工具。如旋转，缩放，切变和翻转，轻易地变换图片的显示效果。
- 多种选择工具。包括矩形、椭圆、自由选择等功能，以适应各种复杂的图片选取环境。
- 对图片中画笔等工具提供支持高品质的反锯齿效果，大幅提高图片中线条的光滑程度。
- 开放插件接口，可以开发新的滤镜或轻松使用他人开发的滤镜，使用 Wine，甚至可以使用 PhotoShop 的插件。

14.5.2 界面简介

GIMP 的界面与常见的软件有所不同。普通程序界面无论如何复杂总是集中在一个窗口内，而 GIMP 的界面却是分为三个独立的窗口，每个窗口虽然功能与其他窗口有联系，但界面与其他窗口独立。这样做的初衷是为了提高操作效率，不过这种怪异的操作方式给第一次使用的用户带来了很大的困扰。其界面如图 14.37 所示。

1. 工具箱窗口

左侧的窗口是工具箱窗口，提供了 33 个按钮来提供相关的功能。如图片区域选择、图片剪切、钢笔、文字、画笔、图章等功能。将鼠标放在图标之上可获得其功能的简单提示。工具箱窗口下面是相关的选项设定功能。这部分是 GIMP 中最重要的部分，用户不能对其进行修改和关闭该部分（除非关闭程序本身）。

工具箱的下面有个黑白色的图标，如图 14.38 所示，可以用来设置前景色、背景色。默认如图所示，是黑色的前景色，白色的背景色。前景色是画笔、铅笔之类的画图工具画出的线条的颜色，背景色是画布的颜色。

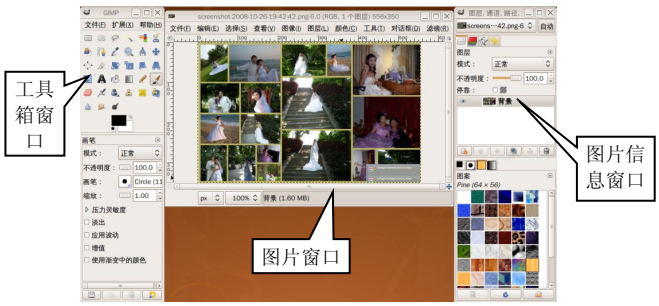


图 14.37　GIMP 主界面

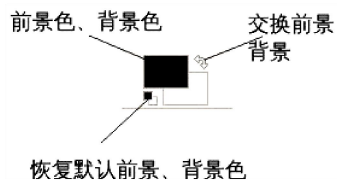


图 14.38　前景、背景设置

图 14.38 中所示重叠的黑白方块分别代表前景色、背景色。单击白色的方块会弹出一个颜色选择对话框，可以选择一种颜色改变前景。单击黑色的方块则是更改背景色。图中右上方的双箭头是用来交换前景、背景色，单击其可以使前景与背景互换。“恢复默认前景、背景色”是用来将图片的前景、背景色设置为默认（黑前景色、白背景色）。

2. 图片窗口

图 14.37 中间的窗口用来显示当前操作的图片。在左侧的工具栏窗口中选择相关的功能后，就可在中间的窗口进行操作，或者也可以在本窗口的菜单中选择相关的功能来对图片进行操作。这个窗口提供了 GIMP 中的所有功能，而工具箱只是提供了部分功能。

该窗口中提供的功能主要是通过菜单来实现的。菜单的功能均可很直观地通过其文字来说明。其功能简单说明如下。

- “图像”菜单主要是设置图像模式、画面大小等。
- “图层”菜单主要是进行与图层相关的操作，如新建、合并等功能。
- “工具”菜单提供了 GIMP 中所有的工具，包括在工具箱中没有显示出来的功能。
- “对话框”是指显示的窗口，可以在该菜单中选择哪些项显示，哪些项不显示。
- “滤镜”是提供了强大的自动化功能用来实现某些特殊的效果。滤镜功能是通过插件来

实现的。

GIMP 安装了许多插件，但用户还是可以自行安装所需的插件。编写插件是普通开发者加入到 GIMP 开发中的最简单的方法，也是演练图像算法的好地方。



GIMP 除了可以使用插件来实现各种功能外，还支持脚本的操作。用户可以自行使用 Script-Fu 语言编写自己的脚本，如果对该语言不是很熟悉，还可以使用 Python 等脚本语言来编写脚本。

在实际的操作之前，简单介绍一下图像与图层，这两个概念是图像处理中很重要的基础知识。

图像是 GIMP 操作的基础，所有的操作对象均是图像。实际上 GIMP 中，图像是放置在画布上的。画布的颜色就是背景色，而画布的大小可以等于图像也可以不等于图像。

图层是稍微高级点的图像处理软件均提供的一项功能。图层就像是一张张透明的薄膜，覆盖在图像上。可以单独对一个图层进行操作，然后将其依次覆盖在背景图案，以达到特定的功能。

3．图片信息窗口

图 14.37 所示界面右侧的窗口显示的内容是图片的相关信息，如图层、通道、历史记录等，还有画笔的一些形状、参数等。

图 14.39 截取了图 14.37 中的图层窗口进行介绍。图 14.39 中有两个图层，当前图层为“背景”层。当前层的显示模式是“正常”，当前层属于不透明的图层。这些均可以单击相关项进行重新设置以达到不同的目的。可以单击下方第一个按钮来增加图层，或者单击最后个按钮来删除图层。图层也只是这个窗口中四个标签页中的其中一页，单击其他标签页可以显示其他窗口。

图 14.40 是图案窗口，显示了系统中所有的图案，单击选中图案后，可以将其拖动到选中区域将该区域用图案填充。同样，也可单击标签转换到其他的窗口。

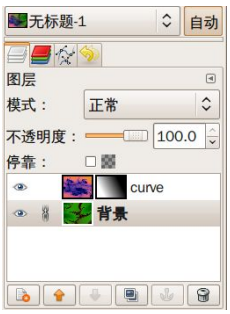


图 14.39　图层窗口

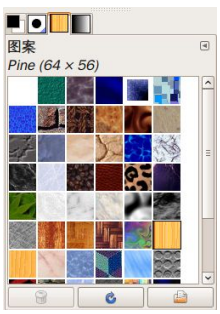


图 14.40　图案窗口

14.5.3　安装插件

系统中提供了许多的插件，但对于许多用户来说，可能还不够强大。这时，就可以自己安装系统中提供的插件。Ubuntu 软件源中有一些 GIMP 的插件，如 gimp-ufraw 是用来读取 RAW 图片，而 gimp-gap 用来将影片转换为 gif。

以 gimp-gap 为例来说明源中提供的插件的安装过程。用命令“sudo apt-get install gimp-gap”安装该插件。安装后，重启 GIMP 可以发现在 GIMP 的界面出现了“Video”菜单项。该插件已经安装成功，可以单击菜单打开相关菜单项实现特定的功能。下面以一个将影片转换为 gif 动画的子来说明其部分功能。

（1）选择“文件”|“新建”命令，新建一个空白图片。这是用来作为图片导出功能的背景，同时做完本步后才会出现 Video 菜单。

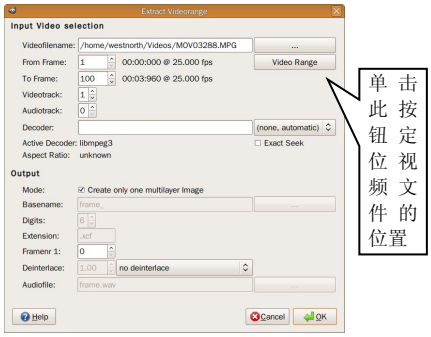


图 14.41　导出影片

（2）选择 Video|Split Video into Frames|Extract Videorange 命令，出现如图 14.41 所示的对话框。

（3）在单击文本框 Videofilename 其后的“…”按钮选择视频文件的位置。

（4）在“From Frame”填上导入视频的开始帧。

（5）在“To Frame”填上截止帧。（可利用“Video Range”查看视频中的帧的预览。

（6）将“Audiotrack 设为 0。

（7）勾选“Create only one multilayer Image”使得输出的格式为多个图片，这样才可以用这导出的多张图片做出 gif 动画。

（8）单击“OK”按钮，保存设置。

（9）现在图层窗口中应该出现了许多从视频中导出的图片。选择“文件”|“另存为”命令，出现保存文件对话框。选择适当的目录，记得要将文件保存为 gif 格式。这样就成功地将视频文件转换为 gif 动画格式。



安装其他用户自己开发的插件同样简单。得到插件后，将其放入“~/.gimp/script”目录中，重启即可在“滤镜”菜单中发现新安装的插件。

14.5.4　示例

前面介绍了 GIMP 的相关知识。但要了解 GIMP 的强大，光了解这些是不够的。本小节用几个简单的例子来展示 GIMP 的一小部分功能。这些功能对于强大的 GIMP 来说，只是沧海一粟，更强大的功能与用法请参阅相关专门讲述 GIMP 的书籍。一般来说，PhotoShop 的教程也可以拿过来在 GIMP 中使用。

1．图像浮雕字

GIMP 的功能与 PhotoShop 的同样强大。而且一个很有用的信息是：大多数 PhotoShop 的教程在 GIMP 中也是适用的。所以本节只简单地在图像中产生出一个浮雕字的效果。

（1）选择“文件”|“新建”命令来创建一个图像文件。出现对话框界面如图 14.42 所示。将宽度设置为 300，高度设置为 200。单击“确定”按钮完成。

（2）选择“文件”|“打开为图层”命令，选择要打开的图像，如“/usr/share/background/space-02.jpg”。然后单击“打开”按钮将图像打开为图层。



这里一定要选择“打开为图层”命令。因为这次打开的文件是作为背景图来使用的。

（3）单击工具箱中的文字按钮（就是一个黑色的“A”字的按钮），开启文字输入功能。这时鼠标变为“I”形状，在图片任意位置单击，弹出文字输入对话框。

（4）在“GIMP 文本编辑器”对话框中文本框内输入“浮雕字”或任何想要变为浮雕效果的文字。

（5）在工具箱中字体设置选项中，将字号设置为 100 号。并用移动工具将文字移到图片中

心位置。现在图层情况应该如图 14.43 所示。



图 14.42　新建图像文件



图 14.43　添加文字后的图层状况

（6）在图层窗口中单击“space-02.jpg”图层，使这成为当前图层。

（7）单击“新建图层”按钮，弹出如图 14.44 所示对话框。将“图层填充类型”选为“白色”，然后单击“确定”完成创建新图层。

（8）现在的图层状况如图 14.45 所示。单击“浮雕字”图层，使之成为当前图层。

（9）选择“图层”|“向下合并”命令，将文字图层与白色背景图层合并。合并后的图层状况如图 14.46 所示。



图 14.44　创建白色背景的新图层



图 14.45　合并前的图层状况



图 14.46　合并后的图层状况

（10）使有文字的图层（本例为“新建图层”）成为当前图层。

（11）选择“滤镜”|“扭曲”|“浮雕”命令，打开如图 14.47 所示的对话框。

（12）将“极角”设置为 10，“仰角”设置为 25，深度设置为 10。单击“确定”按钮完成浮雕效果。

（13）参见图 14.46 在图层模式选择的下拉框中将图层的模式改为“覆盖”，则完成“浮雕字”的创建工作，效果如图 14.48 所示。



图 14.47　浮雕效果设置



图 14.48　最终效果

2．图像互相融入效果

在广告宣传图片中经常可以看到逐渐融入的效果。就是在一张图片中，一种风景（如沙漠）逐渐淡化，而逐渐显示出另一种风景（如湖泊）的效果。这种效果在 GIMP 中也很容易实现。具体实现步骤如下。

（1）选择“文件”|“打开”命令，选择要打开的图像。如“/usr/share/background/space-02.jpg”，然后单击“打开”按钮将图像打开。

（2）新建图层：参见图 14.44 新建一图层，各选项均取默认值，单击“确定”按钮完成新建工作。

（3）选择“滤镜”|“绘制”|“分形探索器”命令，出现如图 14.49 所示对话框。

（4）选择“分形”标签，在右边列表中双击 Leaves 选项，在“实时预览”中出现如图 14.43 所示效果。然后单击“确定”按钮完成分形图形的创建。

（5）现在图层应该如图 14.50 所示。右击名为“新建图层”的图层，选择“添加图层蒙板”菜单项。



图 14.49　分形探索器



图 14.50　图层状况

（6）现在图层状况应如图 14.51 所示。在“新建图层”图层多了一个白色的方框，这就是图层蒙板。

蒙板颜色如果是白色，说明该区域的图片正常显示；如果为黑色，则为不显示；如果是灰色则显示出一种半透明的结果。蒙板只对其所在的图层有效果。现在看上去图像没有发生任何变化，这是因为蒙板是白色，将该层的图片全部显示。

（7）单击“新建图层”层的蒙板（白色方框），使其选中。在工具箱中选择渐变填充工具 。在图像中按住鼠标左键，由左上到右下拖动，然后放开左键。效果图如图 14.52 所示。

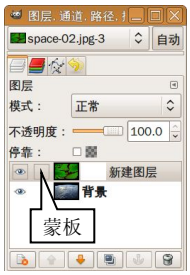


图 14.51　添加图层蒙板后图层状况

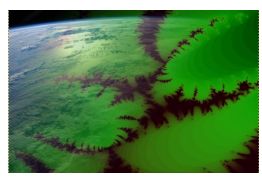


图 14.52　图像融入效果

14.6　矢量图绘制软件

所谓矢量图是指使用线条绘制的各种图形。在文件中存储的信息是函数表示的线条与填充属性，所以矢量图可以无限放大而图像不会有任何的失真。这些特性使得其在某些场合有很重要的应用。本节简单介绍两款矢量图绘制软件，Drawing 主要用来绘制流程图（本书所有流程图均是由其绘制），Inkscape 用来绘制各种复杂的矢量图案。

14.6.1　OOo 办公套件中的绘图软件 OpenOffice.org Drawing

OpenOffice.org Drawing 是一款功能比较强大的矢量图绘制软件，其是 OpenOffice.org 重要组成部分。Draw 的功能如下。

- 绘制流程图：提供了丰富的流程图模块和模块之间连接的方法，可以迅速地供给制出复杂专业的流程图效果。
- 3D 效果绘制：可以将点阵图导入 Draw 中，并且通过“3D 控制器”将平面图片转换为立体效果，如图 14.53 所示。
- 软件中预置了许多模块，直接使用可以减少很多的工作量，如图 14.54 所示。



图 14.53　3D 效果



图 14.54　矢量图模块

- 可以导入所有通用格式的图片，如 GIF、JPEG、PNG 等。
- 可以将图片文件导出为 Flash 文件（swf 格式）。

界面上放置了最常用的功能按钮，方便调用。Draw 的主界面如图 14.55 所示。

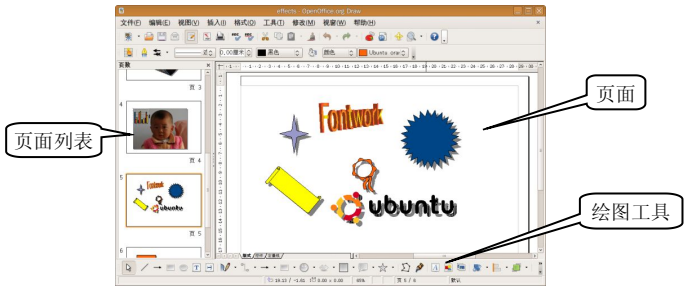


图 14.55　Draw 主界面

菜单栏和工具栏是通用形式，这里就不再赘述。在 Draw 中每一幅图被称为一个页面，左边列出了当前文件中的一些页面。中间显示的是当前的页面，所有的编辑均只对当前页面有效。在当前页面图片下方，是绘图工具，列出了非常有用的一些工具。下面以几个例子来介绍其基本功能。

1．绘制流程图

单击“流程图模块”按钮，会出现一些系统内置的流程图模块，单击第一行第一列的模块：

过程模块。然后在页面中单击，按住不放并将鼠标向右下方拖动，放置“过程”模块。然后用相同的方法再放置几个类似的模块。

单击“连接符”模块，在其中选择一种连接符。将鼠标移动到某一模块中，可以看到模块多了四个圆形的连接点，将鼠标在连接点上单击，然后再移动鼠标到下一个模块的连接点上，再单击，就将这两个模块连接起来了。这种连接符可以自动与模块之间保持连接。如果需要其他功能，如添加文字、添加箭头、插入模块图形等均可以从绘图工具栏中相关按钮找到答案。

2．点阵图处理

Draw 可以导入点阵图，但不会对图片进行诸如色彩、亮度等调整，那些是 GIMP 的工作范围。Draw 可以对图片进行相关的变换，如将其做成 3D 效果等。

将图片做成 3D 效果的步骤如下。

（1）选择“插入”|“图片”|“来自文件”命令，选择一张点阵图（就是普通的数码照片或者网上下载的漂亮图片等，一般后缀为 JPG、PNG、BMP 等），将其插入到 Draw 中。

（2）单击插入的图片，使其选中。

（3）选择“修改”|“转换”|“变为 3 维”命令，可以看到图片已经呈现出初步的 3 维效果，如图 14.56 所示。

看起来效果不太理想，没关系，可以再单击菜单“修改”|“旋转”，然后将鼠标移到到图像上，按住不放，再将鼠标向任意方向拖动，直到满意为止，最后的效果可能如图 14.57 所示。



图 14.56　变为 3 维后的效果



图 14.57　再旋转后的效果



OpenOffice.org Drawing 是 OOo 的一部分。在 OOo 中打开图形绘制功能，实际上就打开了 Drawing。

14.6.2　矢量图绘制软件 Inkscape

Inkscape 的前身是一款称为 Sodipodi 的程序。在 Solipodi 发展到 0.31 版时，开发人员之间产生了分歧，于是一部分人就开始以 Sodipodi 的 0.33 版为基础开始了 Inkscape 项目。发展到现在，Inkscape 已经成为一款功能强大，使用者众多的优秀的矢量绘图软件。Inkscape 可以直接生成 eps 文件，并且 Inkscape 支持 Tex 命令。所以，Inkscape+LaTeX 在书写论文时可以提供很大的帮助。

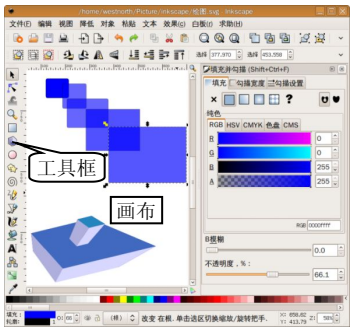


图 14.58　Inkscape 的界面

Ubuntu 的源中有该软件，所以可简单地在终端中运行命令“sudo apt-get install Inkscape”进行安装即可。Inkscape 的界面如图 14.58 所示。启动 Inkscape 可以选择“应用程序”|“图像”|“Inkscape 矢量编辑器”命令或者在终端中输入 Inkscape 来启动这个软件。

Inkscape 的功能强大，不是简单的介绍所能涵盖的，所以仅通过一个例子来简单演示其功能。

1．红十字

本例是一个极其简单的例子。为了演示，仅仅使用了 Inkscape 很少的一部分初级功能，目的是创建一个黄背景上的红十字图案。更为全面的功能介绍可以参见其官方站点。具体步骤如下。

（1）打开 Inkscape 后，选择“文件”|“新建”命令，随意选择其中一个模板（如 Letter），新建一个空白文档。

（2）选择“文件”|“文档属性”命令，打开图 14.59 页面设置对话框。将“默认单位”修改为 px，意思是像素。一般来说，图片长宽比例越接近黄金分割比（0.618），从视觉效果上来看，就越美观。所以我们在这里将图片的宽度设置为 480，而高度设置为 300。

（3）现在新建的文件显示在界面中的一角，单击数字 5 键，使文件占满窗口。

（4）选择“文件”|“文档属性”命令，打开文档属性对话框。选择“网格”标签。

（5）选择矩形网格，然后单击“Grid 新建”按钮，创建一个网格。这时注意看新建的文件，可以看到其中出现了浅蓝色的网格。

（6）将“间隔 X”微调框和“间隔 Y”微调框均设置为 20。

（7）将“主网格线每”微调框项设置为 5。意思是每显示 5 个网格线显示一个主网格线。主网格线较粗，这样可以方便定位及对齐。设置后应如图 14.60 所示。

（8）选择图 14.60 所示对话框“捕捉”标签，打开捕捉对话框，如图 14.61 所示。



图 14.59　文档属性设置



图 14.60　设置网格

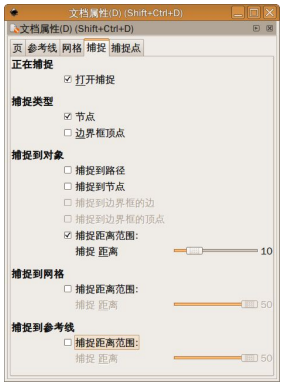


图 14.61　设置捕捉

（9）取消选中“捕捉到网格”、“捕捉到参考线”与“捕捉距离范围”复选框。现在就做完了绘制图形前的准备工作，下一步开始绘制图形。

（10）绘制矩形背景，我们采用填充矩形框的方法。单击主界面中工具框（参见图 14.58 所示）中的“绘制长方形”按钮，或者简单地按下 F4 键，就开始了矩形绘制工作。

（11）开始绘制工作后，将鼠标移到画布中，可以看到鼠标已经变为小十字形状。由于是绘制背景所以本次绘制的矩形要覆盖整个图形。将鼠标移到左上端点处，单击鼠标，由于设置了捕捉功能，只要单击的位置不要太偏，矩形会自动定位在左上角点。然后拖动鼠标到图片右下端点处放开鼠标。现在整个图形均被蓝色的背景覆盖。

（12）现在背景是蓝色，明显不符合我们的目标，我们需要的是黄色的背景。单击图 14.62 所示区域，出现图 14.63 所示对话框，可以在其中设置背景色。将红色（R）的值更改为 255，将绿色（G）的值也更改为 255，将蓝色（B）的值更改为 0，这样就将背景色更改为黄色。

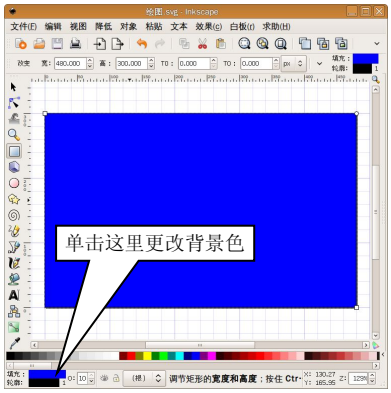


图 14.62　更改填充色

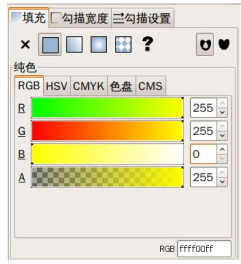


图 14.63　颜色设置

（13）继续选中“绘制长方形”工具，开始绘制中间的十字图形。

（14）仔细观察，就会发现图有五个主网格宽。所以，十字中纵向的一竖应该位于第三个网格。考虑到美观，将一竖的宽度定为 3 个次网格宽。从自左至右第 11 个网格（两个主网格加一个次网格）自上至下第 2 个网格开始绘制竖。考虑到对称，竖的下端离边界也是两个网格。然后按照上 6 左 7 绘制十字的横，具体位置请参照图 14.64 进行绘制。

（15）现在十字绘制好了，但是观察可以发现十字的形状并不理想。因为看起来这是一横加一竖而不是一个十字。这时要做的工作就是将十字的横和竖合并为一个整体。合并前先要将横和竖同时选中，但这时的选中工具不是工具箱中的黑色箭头的工具，要保证选中“绘制长方形”工具，然后单击十字的横，按住 Shift 键，再单击十字的竖，就选中了这两个矩形。此时，仔细观察图形，要保证十字的边上没有 8 个箭头。

（16）现在开始合并，选择 Path|Union 命令，将这一横一竖两个矩形合并为一个十字。现在应该类似于图 14.65 所示效果。

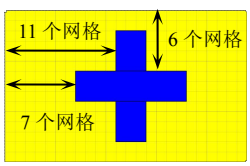


图 14.64　画中间的十字

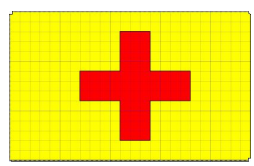


图 14.65　最终效果图



可能中文版没有“路径”|“合并”菜单，这时就需要转到英文环境中运行 Inkscape，方法是打开终端，在其中运行“env LANG=en_US.UTF-8 Inkscape”打开英文界面的 Inkscape 软件，就可以找到菜单“Path”|“Union”。

（17）现在图像成为黄色背景上面有一个蓝色的十字，越来越接近目标了。再单击十字，使其处于选中状态。参照图 14.57 标注单击，然后在颜色设置对话框中将红色（R）设置为 255，将绿（G）与蓝（B）均设置为 0。这样，任务就顺利完成。

2．绘制形状比较复杂的图案

既然 Inkscape 是矢量图处理软件，则矢量图的绘制能力是必不可少的。绘制曲线，比较好的工具就是“贝赛尔曲线”绘制功能。单击工具箱中的“绘制贝赛尔曲线”按钮或者使用组合键“Shift+F6”，就可以打开贝赛尔曲线绘制功能。其具体绘制步骤如下。

（1）打开贝赛尔曲线绘制功能。

（2）将鼠标移到画布中，鼠标形状会变为钢笔尖状。单击画布中任一点，就画下了贝赛尔曲线的第一个点。

（3）移动鼠标到另外任一位置，按下鼠标左键，画下第二个点。同时按住鼠标，然后将鼠标左右移动，会发现两点之间的线变为了曲线，并随着鼠标的移动而变幻形状，这就是贝赛尔曲线。

（4）再按第 2 步所述方法绘制几个点，工具会自动将各点用曲线连起来。

（5）如果最后将鼠标指向绘制的第一个点，则工具会自动将图形封闭。

（6）如果对绘制的曲线不满意，按下 F2 键，鼠标的形状变为尖锐的三角箭头，就可进行修改了。用该箭头单击要修改的曲线，曲线上会出现一些编辑点。单击任意一个编辑点，出现一个控制线，拉动控制线两端的圆形控制按钮，会变幻曲线的形状，如图 14.66 所示。

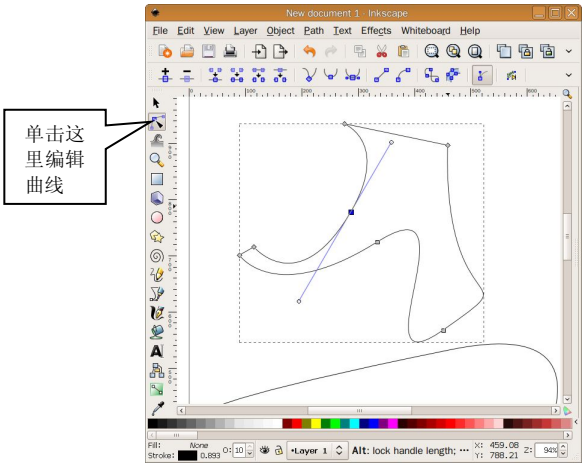


图 14.66　编辑贝赛尔曲线

14.7　CAD 软件 QCad

QCad 是一款 Linux 平台下基于 QT 环境的自由 CAD 软件，其可绘制打开工业标准格式 DXF 文件。相对来说，其绘图功能还算比较强大。不过由于 AutoCAD 的文件 DWG 采用了专利算法，QCad 暂时无法打开，这也从很大程度上限制了其在工程制图界的推广。

14.7.1　QCad 简介

在实际使用 QCad 之前，先来看看 QCad 可以完成什么样的工作，然后考虑是否可以将工作移植到 Qcad 中来。其主要功能如下所示。

- 支持图层：作为一款专业的工程制图软件，图层功能是不可或缺的。QCad 的图层有颜色、线宽、线型几种特性。
- 块功能：可以将常用的一组 CAD 图形组合成一个块，然后使用的时候直接插入，提高效率。
- DXF 输入/输出：支持业界工业标准文件 DXF。

- 丰富的构造工具与修改工具，提供了点、线、弧、圆、椭圆等的构造和修改。
- 尺寸测量与标注功能，可以标注多种表面，直线、圆弧等均可进行标注。
- 丰富的零件库，QCad 提供了几千种 CAD 零件以供使用。

QCad 是款收费软件，不过可以在 Ubuntu 的某些源中免费下载到。方法是配置了源文件后，运行 sudo apt-get install qcad。

QCad 的界面如图 14.67 所示。“工具箱”是常用的工具集合，单击该区的按钮，即可执行相应的功能，如果出现的内容与图示不同，可按下 Esc 键若干次回到这个界面。“制图”窗口是 QCad 中主要的输出界面。“图层”窗口显示当前文件中的图层信息，也可对图层进行管理。“块”窗口显示当前定义的块。“命令行”窗口中显示当前存在执行的命令，用户也可以在这个窗口中输入相关的命令来执行。

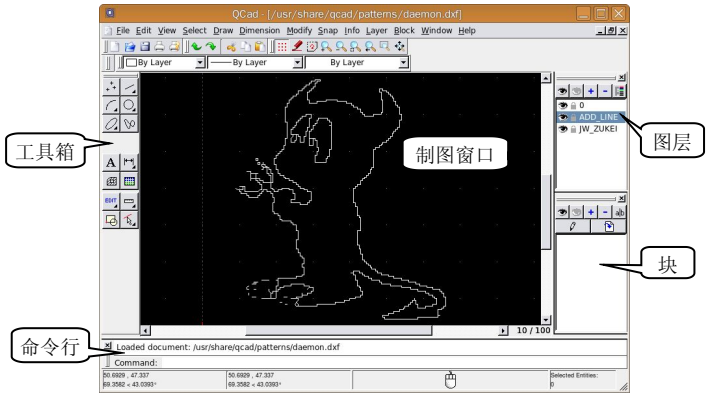


图 14.67　QCad 界面

14.7.2　绘制零件图

CAD 领域，以机械制图最为普遍，所以本节以一个简单的机械零件的例子来简单说明一下 QCad 的相关功能。但这个例子只是介绍了 QCad 的强大功能的一小部分，要了解详细的功能与特色，可以参考其官方网站的用户手册，地址会在本章最后给出。下面是绘制一个零件的过程。

（1）选择“应用程序”|“图像”|“QCad”命令就可打开 QCad。

（2）单击“图层”窗口中的“＋”号，新建图层。出现如图 14.68 所示对话框。

（3）在“Layer Name”（层名）文本框中输入图层名称。先建一个轴图层，所以命名为 axis。

（4）将 Color 下拉列表框设为 Yellow。

（5）将 Width 下拉列表框设为 0.25mm。

（6）将 Line type 下拉列表框设置为 Dash Dot（large）。

（7）单击“OK”按钮结束。

（8）照此方法建立一个名为 main 的图层。Color 设为 Blue、Width 为 0.50mm、Line Type 设为 Continue。

（9）再建一个名为 dim 的图层。Color 设为 Red、Width 为 0.5mm、Line Type 为 Continue。

（10）单击图层窗口中的 Axis 层，使之成为当前图层。因为首先要绘制的是轴线。

（11）单击工具箱中 按钮，然后选择 按钮，选择画线功能。

（12）按空格键启动命令输入功能，在命令行中输入“0,40”（线的起点），按下 Enter 键。然

后输入“130,40”（线的终点），按下 Enter 键，完成轴线的绘制。绘制出的线条如图 14.69 所示。

（13）单击图层窗口的 main 图层，使之成为当前图层。

（14）如果命令行提示是“Specify first point”，则直接输入第一点坐标值。否则重复第 11 步开始画线。

（15）单击空格键，进入命令行模式。输入“5,50”（起点），按下 Enter 键。

（16）提示输入第二点的位置时输入“80,50”，按下 Enter 键结束绘制。

（17）参照上一步，输入第一点坐标“50,65”，按下 Enter 键，输入第二点坐标“120,65”，回车结束绘制。现在，应该如图 14.70 所示。

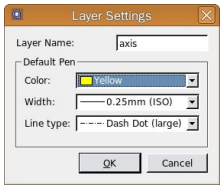


图 14.68　新建图层

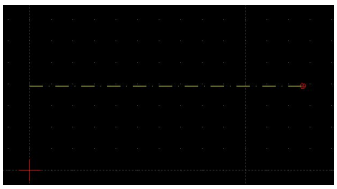


图 14.69　绘制轴线

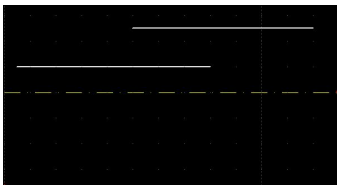


图 14.70　绘制上轮廓线

（18）由于外轮廓线是对称的，所以下轮廓线不用绘制，直接将上轮廓线做一镜像即可。下面是具体的方法：

依次单击工具箱的 EDIT 图标，鼠标指针变为十字中心有一个小方框的形状。单击要镜像的线（就是图 14.70 中两条白色的轮廓线），使其选中。

（19）选中要镜像的线条后，就该选择镜像的对称轴。对称轴是点画线（前面绘制的轴线）。为了更精确地选中该轴线，可以单击，打开端点捕捉功能。现在将鼠标靠近轴线，就会自动定位于该轴线上。在轴线的左端单击一次，选择对称轴的起点，然后在轴线右端单击，选择对称轴的终点。出现如图 14.71 所示的对话框。

（20）选择 keep Original 单选按钮。意思是在做镜像的同时，保留原始的图形。如果选择 Delete Original 单选按钮，就会删除原图形。单击 OK 按钮结束。现在，图形应该如图 14.72 所示。

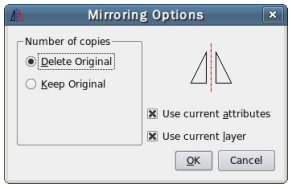


图 14.71　选择镜像方式

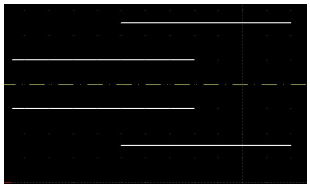


图 14.72　做完镜像后的图像

（21）重复第 11 步，或直接按空格键在命令行中输入 line 命令开始画线。一共需要三条端线，坐标依次是（5,50）→（5,30），（50,65）→（50,15），（120,65）→（120,15）。

（22）依次单击工具箱 EDIT 和 ，开始修剪。

（23）这时命令行的提示是 Select the limiting entity，意思是选择修剪的边界线。参照图 14.73 选择修剪边界线，命令行的提示变为 Select entity to trim。

（24）参照图 14.68 选择两条轮廓线，按下 Enter 键后，完成修剪。现在图形类似如图 14.74 所示的效果。

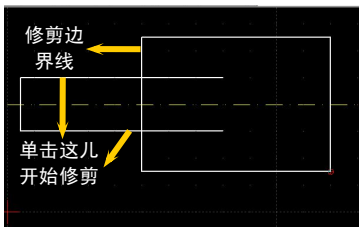


图 14.73　外轮廓线

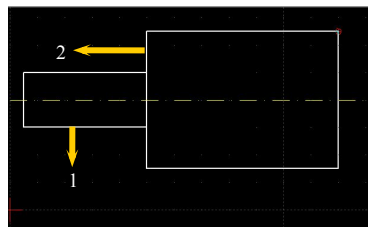


图 14.74　修剪后的图形

（25）由于这仅仅是一个简单的示例，所以图形就绘制结束了。现在开始标注尺寸。单击工具箱中 按钮，打开尺寸标注工具。

（26）单击 ，现在命令行提示 Specify first extension line origin（指定尺寸线的第一点）。在开始标尺寸前，先单击 按钮，打开端点捕捉功能，以准确地标注图形的边界，

（27）将鼠标放置到图 14.74 所示位置“1”处，单击鼠标确定尺寸标注的起点，命令行提示 Specify second extension line origin（指定尺寸线的第二点）。

（28）然后将鼠标移动到位置“2”处，单击鼠标确定尺寸标注的终点。

（29）现在，系统提示“Specify dimesion line location（指定尺寸线放置的位置）”，要单击 按钮将对象捕捉关闭，然后将鼠标移动到高于轮廓的适当位置，单击鼠标，完成标注。

（30）再按照此方法，标注其他尺寸。注意，标注时要先打开端点捕捉功能以准确捕捉到端点的位置，但在放置尺寸线时要将端点捕捉功能取消，不然尺寸线就只能放置在端线上。

（31）再按几次“ESC”键返回到主工具箱，单击工具箱中 按钮，打开尺寸标注工具，单击 按钮按钮，标注其他纵向尺寸。

（32）现在出现一个问题，纵向尺寸是圆的直径，应在其前添加“ø”符号。方法很简单。选择 Modify|Property 命令打开编辑属性功能，然后单击要修改的尺寸标注，出现如图 14.75 所示的对话框。

（33）在对话框 Insert 后的下拉列表框中，选择（Diameter）选项。

（34）单击 Label 后的ø符号，使其选中。

（35）在 Label 后面的文本框中输入原始尺寸，后面还可输入公差数值。具体输入内容如图 14.70 所示。输入结束后，单击 OK 按钮结束。现在就完成了一个简单零件图的绘制、标注过程。成品见图 14.76。

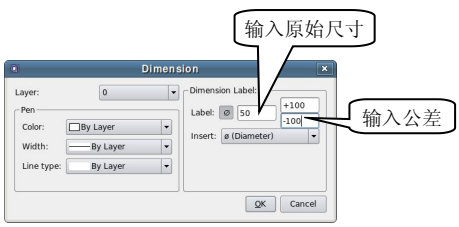


图 14.75　修改尺寸标注属性

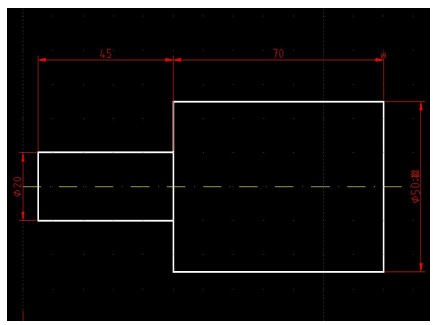


图 14.76　最终效果图



这只是一个演示，实际的制图过程还包括定义模型空间、绘制边框等内容。这里就不再赘述了，有兴趣的读者可以参阅 QCad 的官方中文手册。地址是 http://www.ribbonsoft.com/qcad/manual_reference/zh_CN/。

14.8　小结

本章介绍了 Linux 环境中几种常用的图形图像软件，包含图片浏览软件、图片管理软件、图片处理软件、矢量图绘制软件等。对于普通用户而言，推荐图片管理软件 Picasa，其功能强大、界面美观。而且该软件还是跨平台的软件，在 Windows、Linux、Mac OS 中均可使用。对于图像处理有较复杂需求的用户，则应该仔细阅读 GIMP 的相关章节。由于 GIMP 与 PhotoShop 的操作有很多相似之处，而现在市面上有许多 PhotoShop 的教程，可以参阅这些教程来学习 GIMP。

第 15 章　办 公 软 件

办公软件的使用在企业和日常计算机应用中是很重要的一项内容。其功能强大与否，直接关系到工作中相关任务能否顺利高效地完成，所以办公软件的选择是很重要的。如果没有强大的办公软件，Linux 桌面系统的推广就成为一句空谈。正因为办公软件的重要性，开源世界的各大公司均开发相应的办公套件，如 Sun 公司的 OpenOffice.org、IBM 公司的 Lotus Symphony、国内有金山的 WPS for Linux 版和永中 Office 办公套件。本章主要介绍了 OpenOffice.org 的使用方法，并简单介绍了其他软件的特点及其使用。

15.1　办公软件 OpenOffice.org 简介

OpenOffice.org 是一个网站的网址，同时也是一套软件的名称。作为软件名称时，可以将其简写为 OOo（三个字母“O”，前两个大写，后一个小写）。这就是 Linux 世界中影响最大的办公软件套件 OpenOffice.org。其包括文字处理、电子表格、幻灯制作、数据库等功能，各功能与 Mircrosoft Office 基本对应，也提供了对于 Microsoft Office 的兼容功能。OOo 可以正常打开几乎所有 Microsoft Office 的文件，也可以将文件保存为 Microsoft Office 软件所能打开的格式。这就为与其他用户交换信息提供了便利，因为 Microsoft Office 的市场占有率高达 90%以上。

Ubuntu8.04 附带安装有 OOo 2.4 版，不过 OOo 在 2008 年 10 月发布了 OOo 3.0 版，可去 http://www.openoffice.org 网站下载。OpenOffice.org 3.0 在发布后的一周内下载量达到惊人的 300 万份，在世界引起了广泛的关注。OpenOffice 3.0 版本是其软件版本历史上的重大更新，支持读取 Office Open XML 格式（包括 docx、xlsx、pptx、etc.）文件，并可支持 Mac OS。

OpenOffice.org 包含 6 个组件，但是并不是 6 个软件的简单组合组合，其融合了这 6 大模块，使之成为一个整体。OpenOffice.org 特点如下。

- 系统整体度很高，在一个软件中的设置也会在其他软件中反映出来。如果在一个软件（如 oowrite）中给拼写检查添加了自定义词库，则这些词在其他软件中也可以被系统正常调用。
- 各模块间相同的功能有相同的用法，一次学习，多次使用。
- 可以在一个界面中打开、新建所有的文件格式。例如开启 oowriter 后，同样可以打开电子表格软件，可以将文件保存为电子表格软件格式（格式正确的情况下）。软件会自动调用正确的方法处理用户的操作。
- 发布遵循 LGPL 许可，所有用户均可永久免费使用。
- 可以打开其他人的 Microsoft 文件，如 Word 文本、Excel 电子表格、Powerpoint 幻灯片等。同样也可以将文件保存为 Microsoft Office 可识别的格式，方便与他人的交流。



OpenOffice.org 中的有些功能需要 Java 支持支持，所以应该在运行前安装 Java 6 支持。安装方法是在终端运行命令 sudo apt-get install openoffice. org-java-common。还可安装公式编辑器，命令是 sudo apt-get install openoffice.org-math。而安装数据库软件的命令是 sudo apt-get install openoffice.org-base。

15.2　OOo 文字处理

OOo 文字处理是一款功能强大的软件，可以进行复杂的文档排版、也可以进行图片与文字混合的排版、还可以在文档中绘制美观的流程图等。本节将对这些基本功能进行简单的介绍。

15.2.1　界面简介

OOo 文字处理的界面与其他办公软件的界面没有什么不同。所以使用其他办公软件的用户应该对其界面并不陌生。其主界面如图 15.1 所示。

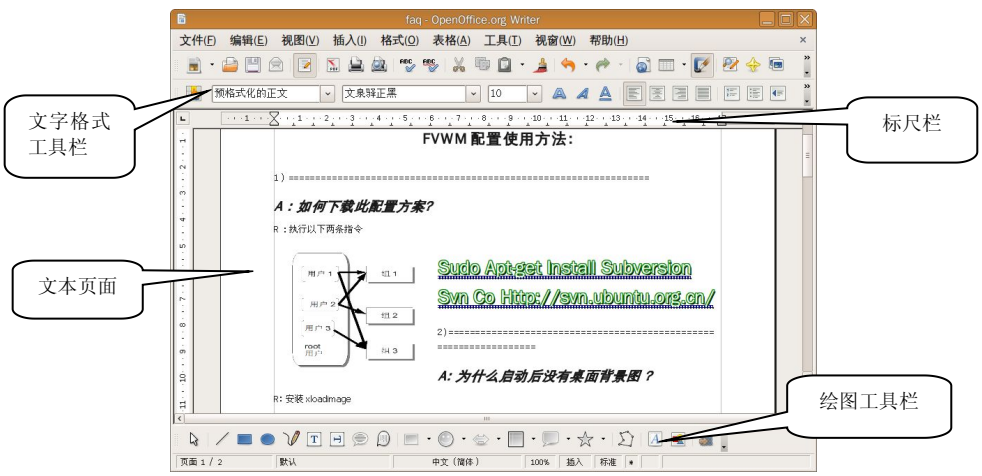


图 15.1　文字处理主界面

文字格式工具栏用来设置文字的格式，如字体、字号、文字的对齐方式等。在设置文本的格式前，要先选中文本，而更改只对选中的文字生效。对要将图中“如何下载此配置方案？”设置为粗体，先在“如”字前单击鼠标，并将鼠标拖动到“案？”处，可以看到文字反向显示，然后单击“文字格式工具栏”中的 A 图标，就可以看到“如何下载此配置方案？”成为了粗体字。

标尺用来控制文字在文本页面的显示范围。标尺是文字显示的界限，一共有三个标尺，分别控制首行缩进、左缩进、右缩进，如图 15.2 所示。

- “首行缩进标尺”控制一段中第一行文字的缩进位置，图中“当前标尺控制的段落”第一行的字符向后缩进了大约 4 个字符，就是由该标尺控制的。
- “左缩进标尺”控制段落中除了第一行的其他行左边界的位置。图中“当前标尺控制的段落”中其他行的左边自什么位置开始就由该标尺控制。
- “右缩进标尺”控制段落中行的右边界。

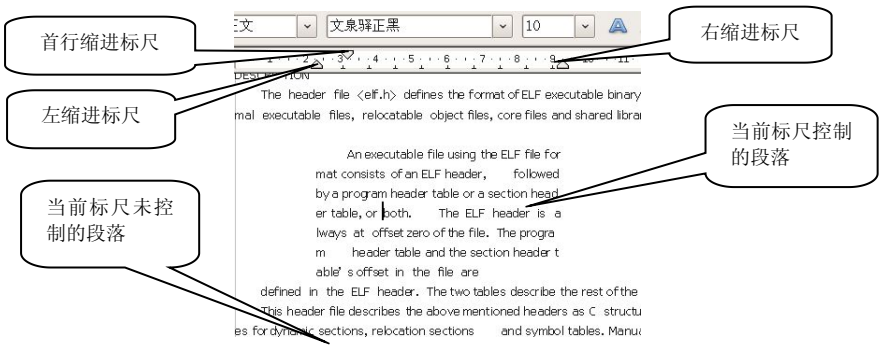


图 15.2　标尺的作用

注 意　标尺控制光标所在的段落的缩进格式，而不仅仅是鼠标选中的内容。也可以用鼠标选中多个段落，让标尺统一控制。

15.2.2　文档排版

OOo 从本质上来说，是一款所见即所得的排版软件，所以熟悉其排版功能应该是很重要的。我们以一首宋朝著名词人柳永的“少年游”为例介绍文档排版。该词的内容是：

少年游　柳永
长安古道马迟迟，高柳乱蝉嘶。夕阳岛外，秋风原上，目断四天垂。
归云一去无踪迹，何处是前期？狎兴生疏，酒徒萧索，不似去年时。

具体的排版步骤如下。

（1）选择“应用程序”|“办公”|“OpenOffice.org 字处理”命令或者在终端中输入 oowriter 打开 OOo 文字处理软件。

（2）打开 OOo 的字处理后，界面类似于图 15.1 所示。当然此时“文本页面”中没有内容。鼠标在“文本页面”中单击，然后将柳永“少年游”的内容输入。输入时先不要理会格式。输入时只管输入，将排版的事情留到以后，实践证明这样做效率会比较高。

（3）做了比较大的修改后，记得要保存文件。虽然 Linux 是一个比较稳定的系统，但也没有人敢保证其桌面环境从不崩溃，所以记得经常保存文件还是必要的。保存文件方法很简单，单击菜单“文件”|“保存”命令，系统会提示要将文件保存为 Microsfoft Office 可识别的格式还是开放文本协议格式（一般保存为 Microsfoft Office 可识别的格式，这样有利于和使用 Windows 的其他人交流。而保存为开放文本协议可能会有稍微好一些的性能优势，可以按照需要选择）。本例选择“否”按钮，保存为 odt 格式。

（4）出现保存文件对话框。选择适当的目录，在“名称”文本框中输入文件名，如 shaonianyou，然后单击“保存”按钮完成保存工作。

（5）一般情况下，词的排版是每行一句，而不是像这样将所有的内容都挤在一行。所以将鼠标移到“少年游”的“游”字后，按下 Enter 键，让“柳永”到下一行。然后照此方法将所有的词均排列为每行一句的形式如图 15.3 所示。

（6）现在文本均位于页面的左端。作为一代风骨的宋词，其应该占据页面的中心位置。按下 Ctrl+A 组合键将文本全部选中，然后拖动左缩进标尺，将其移动到标尺上数字 6.5 处。现在

文本的位置看起来就比较理想了。

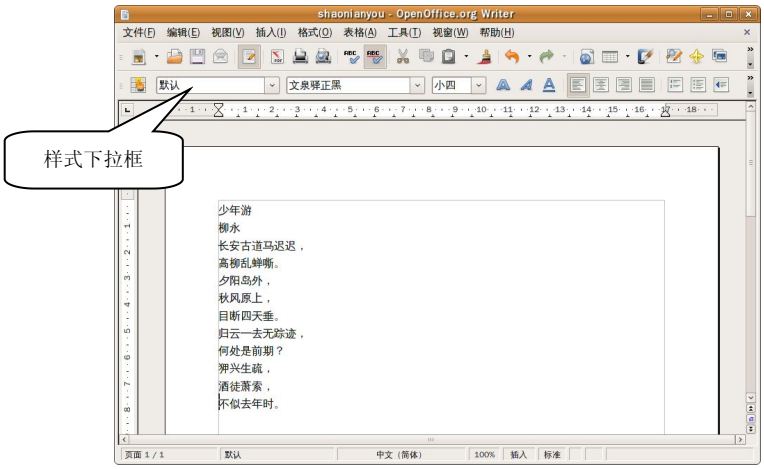


图 15.3　文本的初始格式

（7）在第一行“少年游”中单击鼠标，然后从“文本样式”下拉列表框中选择“标题 1”样式。可以看到，鼠标所在的第一行字号变为“15.1”，而且有了加粗的效果。

（8）现在第一行回到到标尺 0 处，而不再位于 6.5 处了，这是由于“标题 1 样式”定义中标题就应该位于行的左端。再拉动左缩进标尺至 6.5 处就可以了。

（9）将字体再更改为“文泉驿正黑”，完成标题的排版。



文本的样式是一种比较复杂的格式，包括文本的对齐方式、字体、字号等信息。在“文本样式”下拉框里（文本样式下拉框参见图 15.3）预定义的样式只有 4、5 种，要查看详细的样式列表，可按下 F11 键查看全部预定义样式。

（10）现在开始将刚排版好的格式定义为自己的样式。将鼠标放置在刚进行了修改了格式的标题“少年游”中，然后单击 F11 键，弹出如图 15.4 所示对话框。

（11）单击图 15.4 所示位置按钮，会弹出一个菜单，选择“新建”菜单项，出现如图 15.5 所示的对话框。

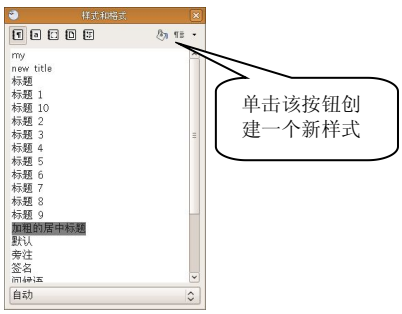


图 15.4　所有的样式表



图 15.5　创建自己的样式

（12）在样式名称文本框中输入自定义的样式的名称，如“加粗的居中标题”。单击“确定”按钮完成。



完成了自己的样式后，第一件事就应该试试，看其是否达到了满意的效果。选中任一行文本（除了标题行，因为样式就是以其为基础创建的），然后从“文本样式”列表中选中刚才创建的“加粗的居中标题”，应该可以看到选中的行变成加粗的 15.1 号字，而且左边界在 6.5 处。

（13）现在标题设置好了，作者名字“柳永”的格式却有点不对。一般来说，作者的姓名应该位于稍微偏右的位置。现在如果直接将其格式设置为“右对齐”也不行，这样会使得名字完全位于右边，偏离下文内容太多。在此采取的解决方案是先将右边界向左拉动一些，然后再将文字设置为右对齐。在“柳永”行单击鼠标，使其成为当前行。

（14）观察文本，发现其最右的文本边界均小于 11，所以将右标尺拉动至 10 处。因为我们不是想让“柳永”靠右太多，只是让其稍微偏右即可。然后，设置文本对齐方式为右对齐。方法是单击工具栏中 按钮。

（15）一般情况，作者的名字字号会稍微小些。用鼠标选中“柳永”，使其反色显示。然后从字体下拉框中将其字体变为“文鼎 PL 简中楷”，这个字体显示出的字体稍微显得小些。

（16）现在版面基本就排好了。但还不够完美，需要做一步的修改。首先，标题与正文是对齐的，而一般情况，从纵向上来看，标题应该位于正文的中间位置。可以采用两种方法来解决，一是拖动标题行的左标尺，向右移动，直到达到满意的效果。二是直接在标题的文字前加空格来修正位置。

词的正文默认情况下是“五号”字，而在许多系统上，文泉驿正黑在小于“小四”号字时显示效果很不好。选中所有的正文，使其反色显示。然后单击字号下拉列表框，将字号更改为“小四”，这样显示效果就好多了。完成的排版效果应该如图 15.6 所示。

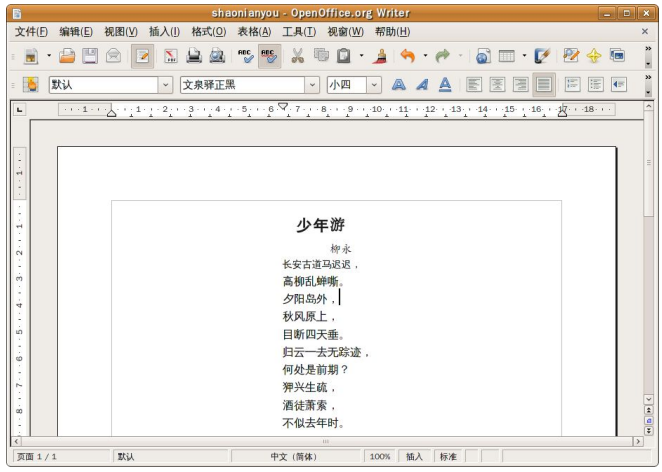


图 15.6　排版结束后的效果

再来解释一下选中与没有选中的状态。参见图 15.7。图中黑色背景的文字：从“我将面向大海”到“要深入我”均为选中状态。其余文字为未选中状态。选中的方法是在“我将面向大海”的“我”前按下鼠标左键，然后拖动鼠标到“深入我”的“我”处，放开鼠标，就可以将其选中。

选中的文字，可以修改其字体、字号、加粗、对齐、颜色、背景等格式选项。而只要鼠标在一段中，如果对其进行标尺操作、对齐方式的操作、样式的更改则均作用于鼠标所在的整个段。

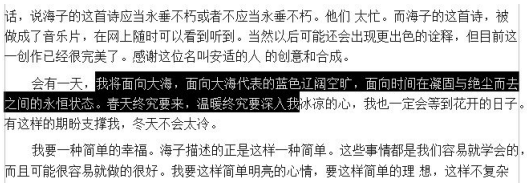


图 15.7　文本的选中状态与未选中状态

15.2.3　文本项目编号

如果在文本处理中，有一些内容意义上是并列的、很难给其进行编号的文本项目，需要以一种比较醒目的方式列举出来，就可以给其进行加上项目编号。选中所有要进行项目编号的文本（一般格式是每行一个项目），然后单击工具栏 按钮，这些文本就被加上了项目编号，如图 15.8 所示。

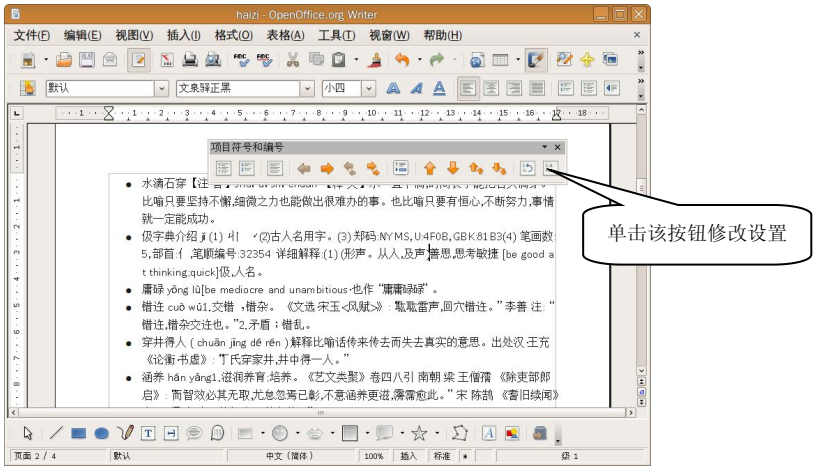


图 15.8　文本项目编号

默认的编号符号是黑色小圆圈，还可以对其进行更改 15.8 所示单击“项目符号和编号”工具条中最右边的按钮。出现如图 15.9 所示编号设置对话框。

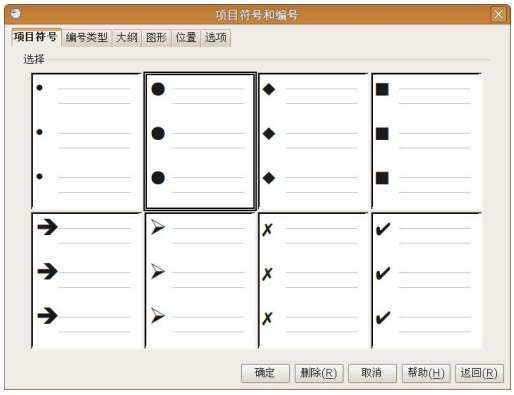


图 15.9　项目符号和编号设置

对话框显示了系统中可用的项目符号样式，单击任一种样式，则该样式会应用于所选文本。如果要对文本进行自动编号，还可单击“编号类型”标签。显示如图 15.10 所示。



图 15.10　编号类型

可以看到，OOo 的文档自动编号功能是比较强的，支持中文也做得比较到位，编号可以支持天干地支和大写中文数字。如果要求编号所用的符号更漂亮些，还可单击“图形”标签页，选择其中漂亮的图形应用于文档中，会给文档增色不少。

15.2.4　图片与文档的混合排版

作为一款功能强大的排版软件，目标就是做出的版面要与报纸看齐。下面以一个复杂的图文混排的版面来演示 OOo 的强大排版功能。目标是实现一个分栏的、标题上有花边的、有图片的版面。下面分别进行介绍。

1．分栏

所谓分栏就是将页面纵向分为几个部分，每个部分可以显示不同的内容。OOo 文字处理的分栏处理功能比较强大。具体分栏的步骤如下所示。

（1）将柳永的词与分栏的内容均输入 OOo 中，作为第一栏中的内容。

（2）选择“格式”|“栏”命令，出现图 15.11 对话框。

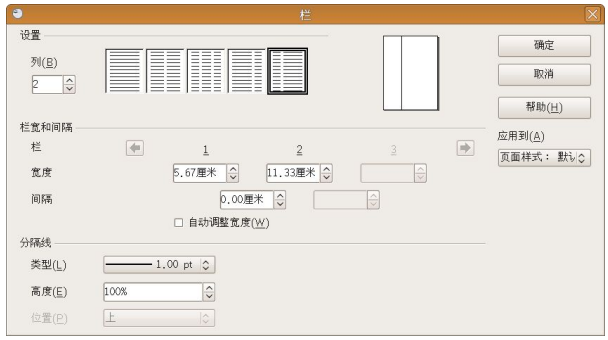


图 15.11　分栏设置

（3）在“列”微调框中输入 2，将页面分为两栏。

（4）在并排排列的 5 个分栏样式中选择最后一个。因为第一栏要显示宋词，而宋词的句子

很短，所以给第一栏留的位置就窄些。

（5）在“分隔线”的“类型”下拉列表框中，将线宽选择为 1.00px。

（6）设置完这些后，单击“确定”按钮完成分栏。

（7）现在再给第二栏输入相应的内容。分栏的效果应该如图 15.12 所示。

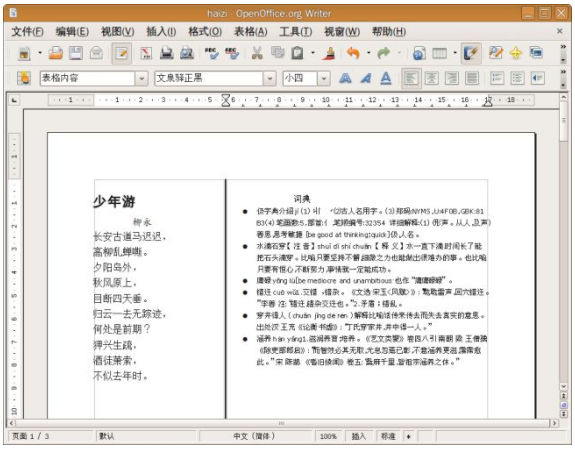


图 15.12　分栏的文本

2．插入图片

很多时候，需要在文档中插入图片来深化主题或是美化版面。在 OOo 文字处理对插入图片的管理还是比较强大的。可以控制图片与文字的放置方式等。下面以一个插入图片的例子来说明其功能，具体步骤如下。

（1）在文档的第 2 栏要插入图片的位置单击鼠标，这在图片插入工序中可以认为是选定了插入位置。

（2）选择“插入”|“图片”|“来自文件”命令，选择一个文件插入文档。

（3）在图片上单击右键，弹出菜单，选择“环绕”|“动态环绕”命令，使得图片周围的文字可以动态地环绕在图片周围。然后，拖动图片到适当的地方。现在的效果应该如图 15.13 所示。

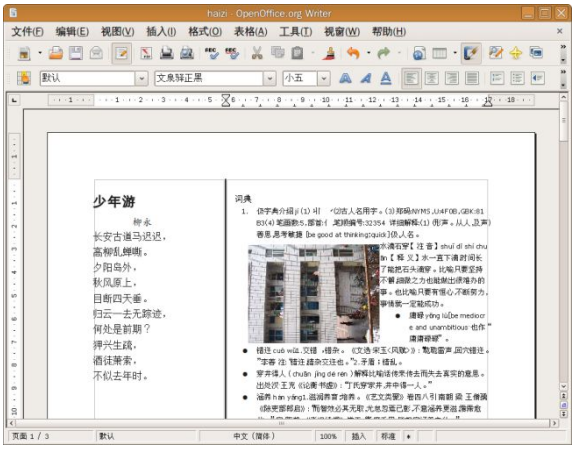


图 15.13　插入图片后的效果

3．花边标题

“花边新闻”原意是指在排版时特意将标题饰以花边的装饰手段。这样的装饰可以起到吸引眼球的作用，在报纸排版中是常用的方法。在 OOo 中实现也比较容易，具体步骤如下所示。

（1）将文档右侧栏的标题“词典”字体大小设置为“小一号字”，对齐方式为中间对齐。



分栏后，每栏相当于一个页面，对齐方式的范围是该栏以内。如中间对齐是指将文字放于该栏的中间。

（2）右击“词典”选项，在弹出的菜单中选择“段落”命令，弹出如图 15.14 对话框，开始设置其花边。

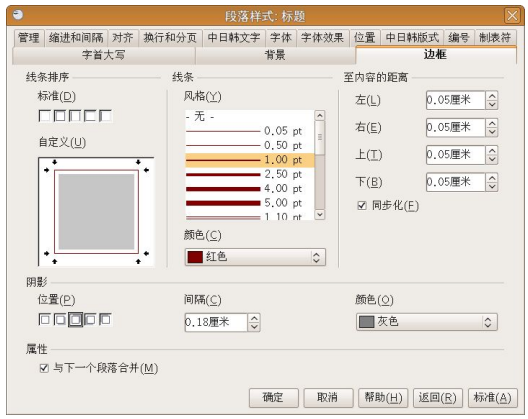


图 15.14　段落边框设置

（3）选择“边框”标签，打开“边框”选择卡。

（4）在“线条排序”栏，单击第二个小方框，意思是选择该段的四周都有线条包围。

（5）在“线条”列表框，单击值为“1.00pt”的线条，意思是包围段落的线条宽度为 1。

（6）选择“颜色”下拉列表框为红色。

（7）在“阴影”栏，单击第三个小方框。选择“间隔”下拉框为 0.18 厘米，颜色为灰色。这项设置后，标题会出现一种阴影效果，增强立体感。

（8）完成以上各项后，单击“确定”按钮完成。

4．彩色背景

段落的装饰中，不仅可以选择段落的边框装饰，还可以选择段落的背景，就是以一种颜色来填充段落的背景。具体步骤如下。

（1）本次操作的对象选择第一栏柳永的词。将柳永的词全部选中并右击，在弹出的快捷菜单中选择“段落”命令，同样弹出图 15.14 对话框，不过这次选择的是“背景”标签。

（2）在“当作”下拉列表框中选择“颜色”选项，对话框中列出了许多颜色，可任意单击满意的颜色，如黄色。然后单击“确定”按钮完成。就可以看到所有选中的段落背景均变为黄色。

如果在“当作”下拉列表框中选择“图形”时，就可以选择一张图片来作为段落的背景了。完成相关设置后，排版的效果可能如图 15.15 所示。

15.2.5　绘制简单的表格

如果在文档处理中需要表格，但是并不需要复杂的数据统计功能，只是简单地将内容列举出来，可以直接在文字处理中绘制表格。方法很简单，单击菜单“表格”|“插入”|“表格”命令，出现如图 15.16 所示对话框。

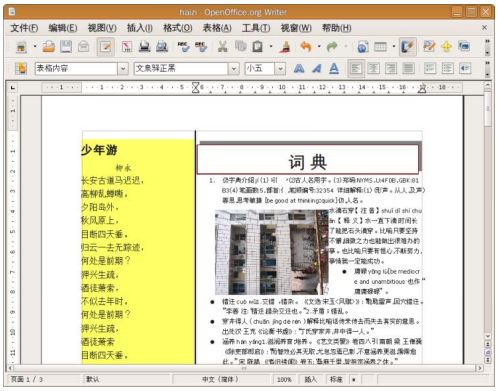


图 15.15　图文混排的效果



图 15.16　插入表格及自动格式

在“表格的大小”列表框中，将“列”下拉列表框中设置为 2，“行”下拉列表框中设置为 2，则建立一个 2 行 2 列的表格。

“选项”中的一些设置也很有用。如果要创建的表格过长，在使用时可能会跨页显示的话。这时可以勾选“标题”项，再选择“重复显示标题”重选框。经此设置，如果表格在显示时被拆分到两页上，会自动显示标题。

而“自动格式”按钮，则可以使表格自动套用某些预先定义的格式。单击其会出现格式对话框。在“格式”列表框中选择一种格式，如“默认”，单击“确认”按钮完成。创建的表格应该如图 15.17 所示。

表格的编辑功能，如添加行，添加列，删除行，删除列等均可在图 15.17 所示“表格”工具栏中找到，就不再详细介绍了。这里只简单介绍一格的求和功能。一般来说，表格使用中求和功能是最为常用的。所以字处理中的表格也有这个唯一的计算功能。

（1）单击工具栏中的添加行按钮 5 次，添加 5 行，在除了最后一行的其他各行的第一列依次填入数字。如填入 1、2、3、4、5。

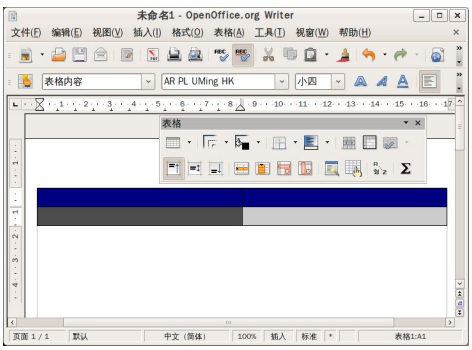


图 15.17　新建表格

（2）单击最后一行的第一列，使该格成为存放求和结果的位置，然后单击工具栏中的求和符号，如图 15.18 所示。

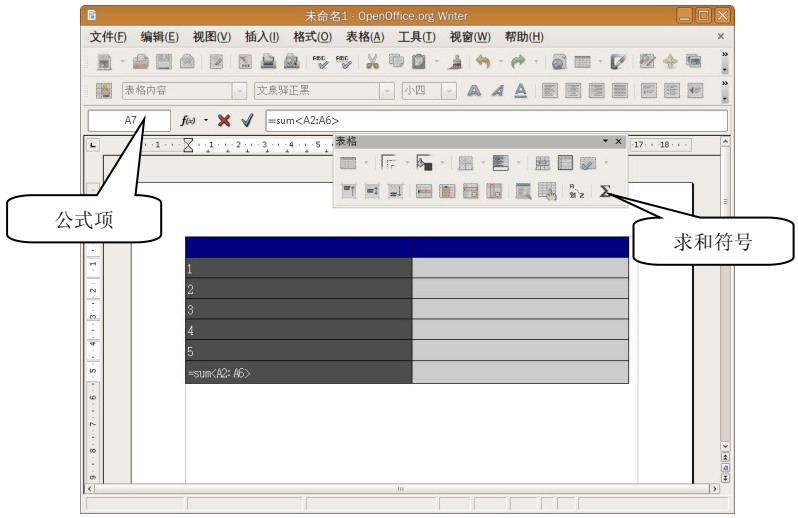


图 15.18　表格求和

（3）现在多了一个公式项，这时可以在其后的文本框中输入要求和的表格范围。表格的列由左至右编号依次是“A、B、C、…”，而行由上至下编号则为“1、2、3、…”，格式类似 sum<A2:A6>。

也可以用鼠标拖动选择要计算的表格范围。本例中，可先在 A2 格单击，然后按住鼠标拖动至 A6 格。可以看到，随着鼠标的选择格不同，最后一行第一列显示的内容也会不同。选择好表格范围后，按回车键确认，就会看到最后一行第一列的内容成为前面几格的数字相加之和“15”。



要绘制比较复杂的表格还是应该使用专门的表格软件。这里只是演示了文字处理软件的绘制表格的功能，可供特殊需要时使用。

15.3　OOo 电子表格

OOo 的电子表格软件可以用来制作复杂格式的电子表格、可以导入格式正确的纯文本文件、可以用其强大的公式功能来对数据进行统计分析、还可以制作美观专业的图表。这些功能在办公应用中是很常见的，所以应该掌握。

15.3.1　软件界面

打开电子表格的方式是选择“应用程序”|“办公”|“OOo 电子表格”命令或者在终端中输入 oocalc 就可以打开了。其界面如图 15.19 所示。

界面的主体是一个个的格子，可以在格子中输入文字，数据甚至插入图像。至于工具栏、菜单对读者来说应该很熟悉了，和其他 OOo 软件的界面的显著区别是有一个“公式栏”，在公式栏中，可以输入公式来完成一些复杂的运算。

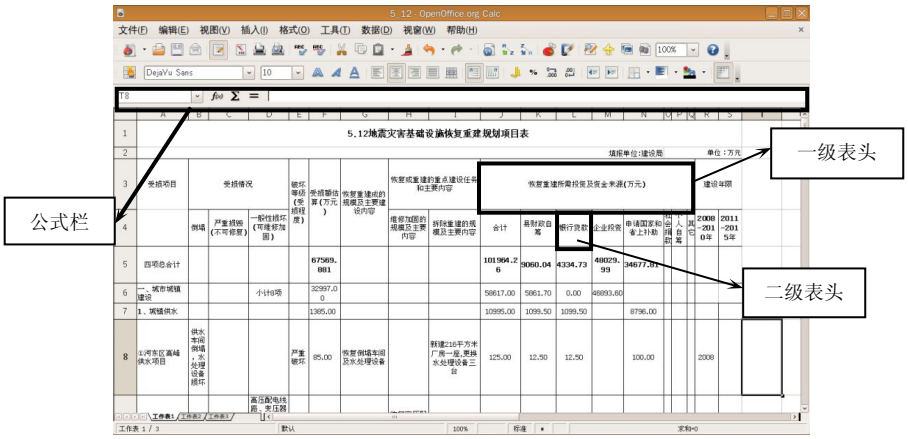


图 15.19　OOo 电子表格软件界面

15.3.2　制作复杂格式的电子表格

图 15.19 所示打开的表格格式很复杂。但是用 OOo 电子表格来制作其实也很简单。制作复杂的表格，要在制作前先分析其格式，而不宜盲目地动手，盲目动手只会事倍功半。然后根据需要对表格进行合并、拆分等操作。一些关于绘制复杂表格的介绍如下。

1．分析格式

分析表格格式是对表格有个大体的规划，这样在制作表格时就可以有的放矢了。这是制作表格之前应该做的第一件事。分析表格，最重要的内容是看表头。因为表头是最为复杂的部分。下面是分析表头的一般步骤。

（1）先看表头。看表头时，一般应该先数一共有多少列。这个列数包含了所有的分项的列。如图 15.19 所示从左到右一共有 19 列，所以初步将 A 到 S 列作为表格的范围。

（2）再看表头的行，图 15.19 所示表头有两行。也就是说只包含了一级表头与二级表头两种。

（3）一般来说标题与副标题要占两行，所以表头应从从第 3 行开始。

（4）然后就可从第 5 行开始输入数据了。（第 1 行为标题，2 行为副标题，3 行为一级表头，4 行为二级表头。）

2．输入数据

在表格中输入数据应该是最基本也最简单的表格操作技巧了。方法很简单，鼠标单击任一单元格，就可在该格内输入。一格输入完成，按“回车”键将输入焦点移动到下一行，或者按“TAB”键到下一列。可参照图 15.20 在电子表格中输入数据。

如果是两级表头，将一级表头输入到第一行，二级表头输入第二行。在输入二级表头的表格第一行不能再输入其他的一级表头。

3．合并单元格

其实图 15.19 所示复杂表格均是使用合并单元格形成的。内容全部输入后，就需要对其格式进行设置，设置格式的主要方法是合并单元格。

合并单元格的步骤如下。

（1）用鼠标单击 J1 格，然后拖动鼠标到 Q1 格，使其反色显示，如图 15.21 所示。

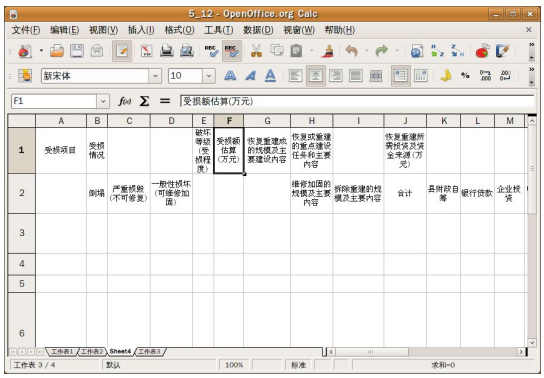


图 15.20　输入格式

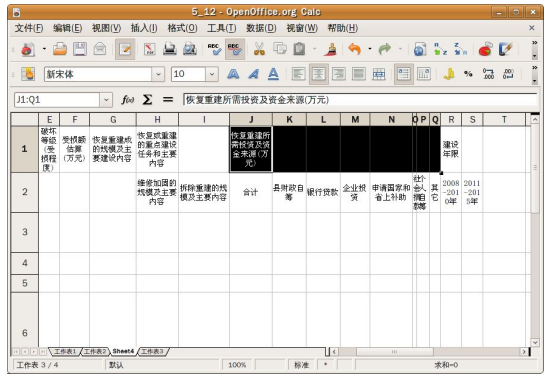


图 15.21　选中单元格

（2）单击工具栏“合并单元格”按钮（如图 15.22 所示），将其合并。

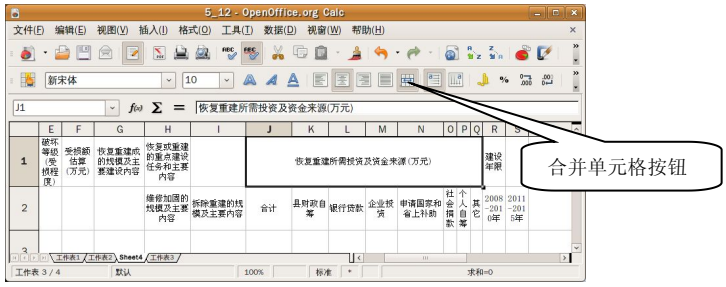


图 15.22　合并单元格按钮示意

同时将其他一级表头合并。合并方法参照图 15.21 与图 15.22 所示。

15.3.3　调整显示格式

由于表格内容的不同，所以各行需要的宽度及各列需要的行高可能均不相同，所以需要进行宽度等方面的调整。

1．调整表格宽度

将鼠标移动到表格列号中间，光标形状会变为双箭头形式。单击鼠标，然后向左右拉动，就可以看到鼠标前面的列的宽度发生了变化。如果选中了多列的话，拖动鼠标会将所有选中的表格宽度拉大、缩小。

还可以将多列选中，然后在列号上右击，在弹出的快捷菜单中选择“列宽”命令，在弹出的对话框中输入数字调整选中的列的宽度，如 15.23 所示。如果在菜单中选择“自动列宽”项，可以让软件自动选择合适的列宽，非常适于那些不想一项项设置列宽的用户。

2．调整表格高度

方法与调整宽度类似，鼠标在表格的行号之间单击，拖动鼠标，就可以更改表格的高度。或者选择多行，然后拖动鼠标；或者在行号上单击右键，选择“行高”在对话框中输入行高的具体数值。

3．调整对齐方式

图 15.21 所示的“合并单元格”功能其实还有一个附带的功能，就是将合并后的单元格文

本居中对齐。所以合并后的单元格文本是居中对齐的，但是其他没有合并的单元格对齐方式就不能让人满意。下面是设置单元格对齐方式的步骤。

（1）将要更改对齐方式的单元格选中。

（2）在选中的单元格上右击，在弹出的快捷菜单中选择“单元格格式”命令，弹出如图 15.24 所示对话框。

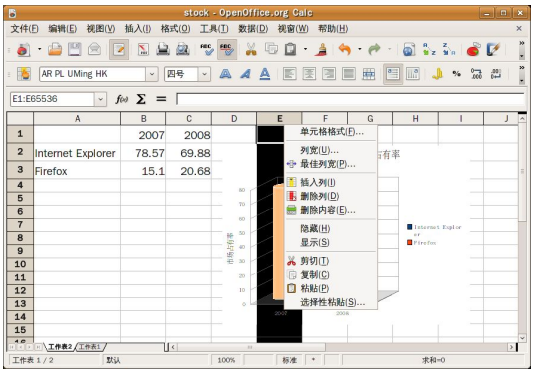


图 15.23　设置列宽



图 15.24　设置对齐方式

（3）其中“文字对齐”栏可以选择单元格在水平方向和垂直方向的对齐格式。如果需要让单元格中的文本完全处于单元格的中心，可以将“水平”和“垂直”对齐方式均变为“居中”。如果需要让文字倾斜一定的角度，可以在“文字方向”栏中转动圆形转盘，使单元格的文字依此旋转。特别地，如果要让单元格中的文字竖排，可以选中“垂直堆积”（这里翻译得并不准确），然后选择“使用亚洲语言版式”复选框，这样单元格中的文本就会像中国的古书一样，成为竖排版的方式。

15.3.4　绘制表格线

现在已经将格式设置成功。但还缺少最后一项，就是表格线。在没有添加表格线时，屏幕显示可以看到表格周围有表格线，实际打印出来后是没有表格线的。添加表格线的方法很简单，具体参照如下步骤。

（1）将要添加表格线的单元格选中，如图 15.25 所示。

（2）在选中的区域中单击右键，在弹出的菜单中选择“单元格格式”，出现如图 15.26 所示对话框，可以在其中设置表格线。

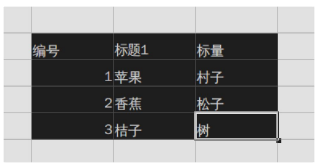


图 15.25　选中要添加表格线的表格



图 15.26　表格边框

图 15.26 中表格线的绘制功能可以看起来不太直观，下面用图 15.25 参照对比讲解一下。



表格由两种线条组成，外边框线和表格内框线。外边框线是表格最外的线，共有上、下、左、右四个边框线。表格内框线只有两种：表格内水平线、表格内垂直线。这几种线可参照图 15.26 中“自定义”框来加深理解。单击“自定义”框相应的线条可以设置对应的表格线条。

（3）“线条排序”这项看起来有些奇怪，其实其英文原文是“Line Arrangement”，翻译成线条布置可能更为确切些。在该栏的“标准”选择项有五个小方框，由左至右依次是：

- 不采用边框：默认的选项。意指不在选定的单元格中绘制任何边框线，这项功能可以用来取消已绘制的边框线；
- 采用外边框：只绘制外边框线（上、下、左、右）；
- 采用边框和水平线：绘制外边框线和内边框线中的水平线，不绘制垂直线；
- 采用边框和全部内框线：绘制所有外边框线和内框线。最常用的选项；
- 采用外框线，不改动内框线：只绘制外边框线，不改动内边框线。可以用于要单独设置外边框线的情况。



图 15.27　设置结束后表格的打印效果图（局部）

（4）在“线条”栏中选择“风格”为 1.00pt。

（5）单击“标准”项中第四项“采用边框和全部内框线”，将所有的线条均设置为 1pt 宽。

（6）单击图 15.26 中内框水平线，使其选中。

（7）在“线条”栏中选择“风格”为 0.50pt，现在水平内框线宽度设置为 0.5pt。

（8）单击垂直内框线，将其也设置为 0.5pt。绘制了线条的表格效果如图 15.27 所示。

15.3.5　导入文本文件

有时，我们得到的文件是有格式的纯文本文件，如用空格隔开的一些数据。像下面这样：

```
    日期 开盘价    最高价    最低价    收盘价    Avg Vol 已调整收盘价*
    08 年 10 月 27 日    1，809.22    1，809.22    1，664.93    1，771.82    4 ，  294 ，
967，2001，771.82
    08 年 10 月 20 日    1，924.18    1，996.66    1，825.53    1，839.62    4 ，  294 ，
967，2001，839.62
    08 年 10 月 13 日    1，978.56    2，146.73    1，901.50    1，930.65    4 ，  166 ，
285，3001，930.65
    08 年 10 月 6 日     2，267.39    2，267.39    1，963.18    2，000.57    4 ，  294 ，
967，2002，000.57
    08 年 9 月 22 日     2，241.72    2，333.28    2，115.96    2，293.78    4 ，  294 ，
967，2002，293.78
```

这是 2008 年上海证券交易所综合指数的历史记录，包含成交量和成交价等数据，可以从 http://finance.cn.yahoo.com/q/hp?s=000001.SS&c=2000&a=00&b=4&f=2008&d=09&e=30&g=w 得到。下载的文件后缀应该是 csv。

导入的具体步骤如下所示。

（1）选择 OOo 电子表格软件菜单“文件”|“打开”命令，打开如图 15.28 对话框。

（2）找到下载的以 csv 为后缀的数据文件，单击“打开”按钮，出现图 15.29 格式设置对话框。

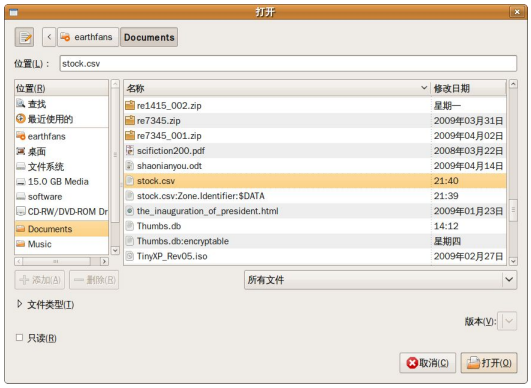


图 15.28　导入文件



图 15.29　导入格式设置

（3）“字符集”选择文本文件的编码，Ubuntu 默认的编码是 UTF-8，所以如果在 Ubuntu 中生成的文件，选择 UTF-8 是不错的选择。如果编码选择正确，在“导入预览”的文字就是可以识别的，如果不正确，则会出现乱码。如果不想导入文本中前 3 行的数据，可以将“开始的行数”设置为 3，否则保持为默认的 1 即可。

“分隔选项”是指数据文件中文本分隔的符号，这个可以根据实际情况选择，一般文件是用空格或制表符来分隔各数据项。选择正确的分隔符，“导入预览区”会显示正确的导入内容。设置结束后，单击“确定”按钮完成导入。导入后的效果如图 15.30 所示。

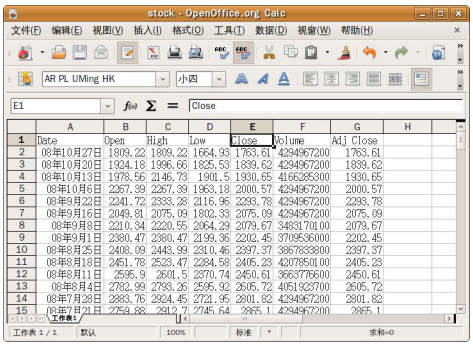


图 15.30　导入的数据

15.3.6　图表制作

向表格中插入图表会从很大程度上增强文档的表现力，可以很直观地观察到复杂数据之间的关系、趋势等。所以在数据统计及分析类的电子表格文件中，会大量使用图表。OOo 电子表格可以制作出多种效果的专业级图表，其支持的图表类型如图 15.31 所示。

当然，以表格中数据为基础生成图表也很简单。只需要两步，以上例中导入的股票历史记录文件为例，介绍如何插入图表。下面是具体步骤。

（1）单击“日期”的列号“A”，按住“Ctrl”，再单击“收盘价”的列号“E”，将这两列选中。

（2）单击工具栏中图表按钮，弹出对话框，单击对话框中“确定”按钮，完成图表创建工作。就这么简单。生成的图表类似于图 15.32 所示。



图 15.31　OOo 提供的图表类型

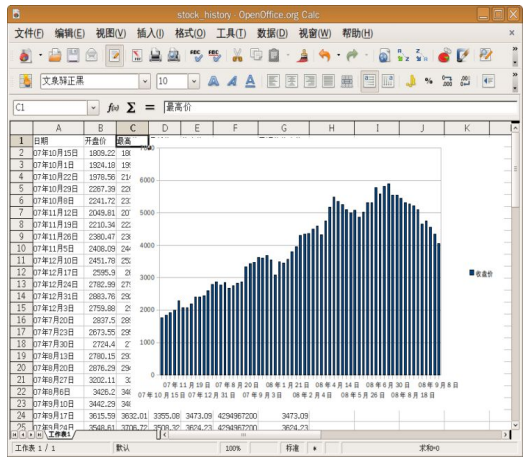


图 15.32　图表

这个例子虽然易于操作，但其实对于 OOo 的图表制表介绍还是过于简略了些。下面介绍一个数据很少，但很完整的制作图表的例子。步骤如下。

（1）在电子表格中输入 2007 年、2008 年 IE 与 Friefox 的市场份额，如图 15.33 所示：

（2）将其全部选中。

（3）单击图表按钮，出现如图 15.34 所示对话框。

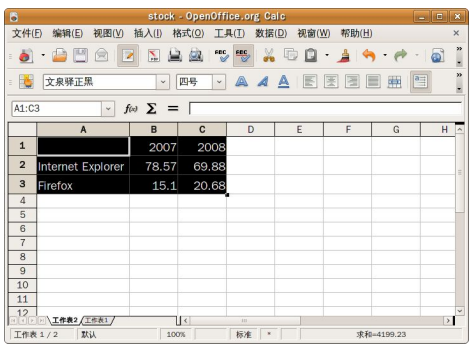


图 15.33　IE 与 Firefox 在 2007 年、2008 年市场份额表



图 15.34　选择三维外观

（4）在“图表类型”列表框中选择“柱形图”。

（5）选择右边的“三维外观”复选框。

（6）在“形状”列表框里选“圆柱体”选项，单击“继续”按钮到下一步，如图 15.35 所示。

（7）选中“行中的数据序列”单选框，图表会先显示第一列的内容，然后显示第二列的内容。以图 15.33 为例，先显示 2007 年 IE 和 Firefox 的占有率，再显示 2008 年的。如果选择“列中的数据序列”，则会先显示 IE 在 2007 年和 2008 年占有率，然后显示 FireFox 的。



图 15.35　选择数据区域

（8）选择“第一行作为标签”和“第一列作为标签”复选框，这是由于在选择数据区域时将标题的内容也包括了，这样做的好处是使软件生成图表时自动加上标签。单击“继续”按钮到下一步。

（9）“数据序列”可以接受默认值，单击“继续”按钮到下一步，如图 15.36 所示。

（10）在“主标题”文本框中输入图表的主题。

（11）在“副标题”文本框输入制作人的名字。

（12）在“_轴标题”文本框中输入该轴的说明“市场占有率”。

（13）将图例显示位置改变到“下”。

（14）单击“完成”按钮结束设置。现在，就可以看到一个美观的图表已经插入进表格中了。效果如图 15.37 所示。



图 15.36　设置图表元素

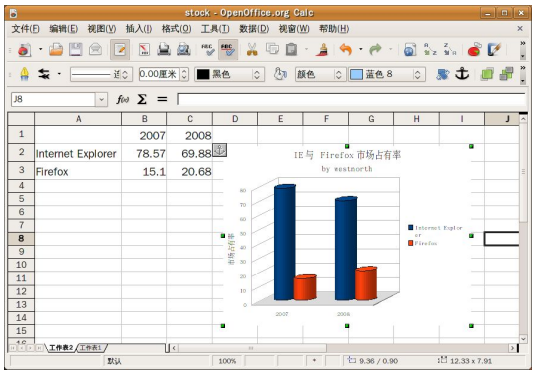


图 15.37　图表完成后效果

15.3.7　公式计算

OOo 的电子表格功能很重要的一项就是进行自动计算，使用公式的自动计算功能可以很大程度上提高工作效率。公式的使用很简单，在单元格中输入内容时，先输入一个英文的“=”就开始了公式输入功能，在公式中还可调用系统中预置的大量的函数。

本例中，我们以一个计算身体质量指数（BMI，Body　Mass　Index）的公式来演示其公式计算功能。BMI 等于体重除以身高的平方，男性的 BMI 等于 22，则说明符合健康标准。年轻女性的 BMI 应该在 20 到 21 之间，如果 BMI 小于 18.5 说明营养不良。具体步骤如下所示。

（1）在表格中输入图 15.38 的内容。

（2）输入文字后，鼠标单击 B5 格，就是图中有黑框的单元格。检查现在的输入状态，如

果中文输入法开启的话，将其关闭，保证输入处于英文状态。然后在 B5 格中输入“=”，开始公式输入。

（3）输入“=”后，鼠标单击 B4 格，意思是引用 B4 格，这时 B5 格中应该出现一个 B4 的字样。

（4）输入除号“/”。

（5）输入左括号“(”。

（6）单击 B3 格，引用其值。

（7）输入乘号“*”。

（8）单击 B3 格，引用其值。

（9）输入右括号“)”。现在 B5 格的显示内容应该类似于图 15.39 所示。

（10）单击回车键完成公式输入。B5 格会显示除 0 错误。这是由于除数 B3 格没有输入数据的原因，在其中输入数据就会等到正确的结果。

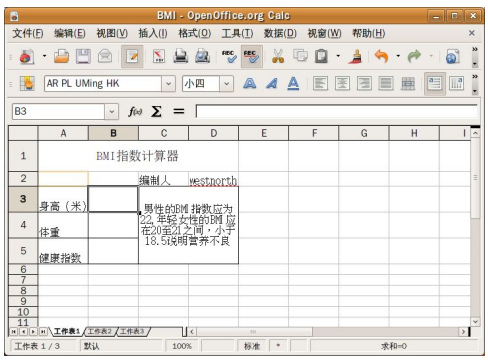


图 15.38　公式计算前输入的内容

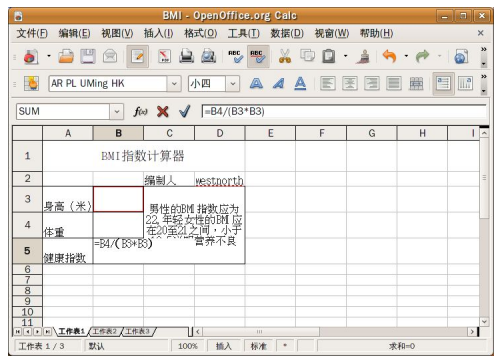


图 15.39　公式输入结束时表格内容

现在，公式输入完成，可以在 B3 格中输入身高，在 B4 格输入体重，检测一下自己的 BMI 看是否符合世界卫生组织的标准。

公式计算功能很强大，而 OOo 电子表格中内置的函数更是极为丰富，现在来简单介绍一下。将鼠标放置于要进行函数计算的单元格中，单击工具栏 $f(x)$ 按钮，打开函数选项，如图 15.40 所示。



图 15.40　函数向导

在分类下拉列表框中，有数学、日期与时间、统计、财务等多种分类，选择后，在下面的

列表框中会出现该分类的所有函数，方便调用。图示是一个数学类中的求绝对值的函数。

现在既可以直接在“函数参数输入框”中输入具体的数值，计算该值的函数值。也可以输入单元格的地址，如 A5 等，让函数计算该单元的函数值。如果对单元格的内容记忆不准确，可以单击“选择单元格”按钮，然后选择要进行函数计算的单元格，进行计算。

来看一个具体的使用函数求自然对数的例子，理解了这个例子，就可以将 OOo 电子表格当做一个高级计算器使用。具体步骤如下。

（1）选用 A1 格作为参数值的存放格（可以任意选取）。在其中输入参数值，如 12。

（2）选择函数结果的存放格，如 A2。方法是单击 A2 格，使其选中。

（3）单击工具栏 $f(x)$ 按钮，出现 15.41 对话框。

（4）在其“分类”中选择“数学”。

（5）在“函数”列表框中选择 LN。

（6）单击“选择单元格”按钮，这时会回到表格中。

（7）单击 A1 格，回车确定选择，会回到图 15.35 所示的对话框，单击“确定”按钮，完成函数。

（8）做完以上 7 步，就会看到 A2 格的内容变为 ln（15）的值 2.48。

15.4　OOo 演示文稿

在向客户展示一个商业计划、或者做一个专题演讲等等这样的场合，如果配合一个精彩的演示文稿，会给你的商业计划或演讲增色不少。用 OOo 演示文稿软件就可以完成这项工作，制作出漂亮的演示效果。

1．创建演示文件向导

选择“应用程序”|“办公”|“OOo 演示文稿”命令或者在命令行输入 ooimpress 就可以打开该软件。第一次运行时，会打开一个创建演示文件向导，如图 15.41 所示。



图 15.41　文档向导 1

具体步骤如下。

（1）在类型中选择“来自模板”复选框。

（2）在“演示文稿”列表中选择“Human Ubuntu”，单击“继续”按钮到下一步。

（3）在幻灯片设计中，各项均取默认值，单击“继续”按钮到下一步。

（4）本步是选择幻灯片切换方式，将“效果”设置为“随机切换”，而“速度”设置为“快速”。 单击“继续”按钮到下一步，如图 15.42 所示。



图 15.42　设置幻灯的基本信息

（5）本步是设置幻灯的基本信息，在各文本框中输入相关信息，这些信息将在每页幻灯片中显示出来。单击“继续”按钮到下一步。

（6）现在创建基本完成，单击“创建”按钮完成，就来到了 impress 的主界面，如图 15.43 所示。

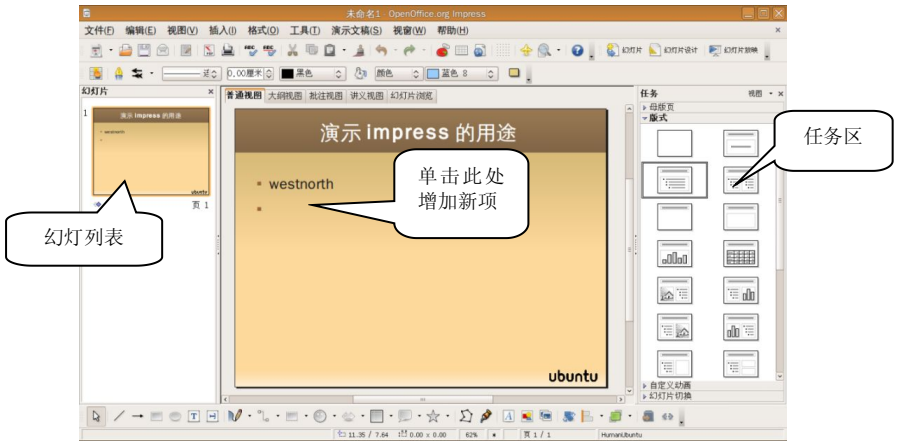


图 15.43　主界面

如图 15.44 所示，“幻灯列表”中显示了当前演示文档中的所有幻灯片，在该区单击右键，也可以新建幻灯片。“任务区”是幻灯制作中的主要功能区域，该区域可以完成幻灯样式设计，切换效果设置等工作。中间的显示区显示了当前正在编辑的幻灯片，还可以单击页面上的标签切换至其他视图形式，如“大纲视图”、“批注视图”等。

2．创建一张幻灯片

单击显示区中的空白的“ •”行，输入文字（如图 15.43 所示），然后按下 Enter 键，增加新的一项列表，继续输入内容如图 15.44 所示。

3．创建有图片的幻灯片

图 15.44 所示的是一张列表格版式的幻灯片，适于列举意义并列的内容。当然还可以增加其他版式的幻灯片以展示不同的内容，如展示一张图片就可以使用有图片的幻灯。

在幻灯中插入图片，有许多方法，我们采取最简单的一个方法，步骤如下。

（1）新建一张幻灯，单击“任务”区的“版式”栏中的“标题、剪贴画、文字”按钮，将幻灯版式设置为带图片的形式。

（2）根据提示信息，双击幻灯片的左侧区域，选择一个图片单击“确定”按钮将图片插入。

（3）在“单击插入标题”和“单击鼠标新增大纲”处分别单击鼠标，输入标题和对于图片的说明文字。就完成了带有图片的幻灯的制作。效果图如图 15.45 所示。

这是一个图片在左侧，而说明文字在右侧的版式，在版式中还有一个是图片在右侧，而说明文字在左侧的版式，可以按照需要选择使用。

4．在幻灯中绘制表格

在版式中选择“标题、电子表格”项，使用带有表格的幻灯版式。按照幻灯中的提示，在幻灯中间区域双击鼠标，打开电子表格。实际上此时打开的就是 OOo 的电子表格软件，只不过其嵌入在幻灯中而已。一个插入表格可能的结果如图 15.46 所示。



图 15.44　第一张幻灯片

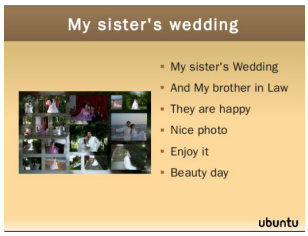


图 15.45　有图片的幻灯片图



图 15.46　有电子表格的幻灯片

5．设置其中一个元素的动画效果

幻灯设计中，可以让幻灯片中一个元素，如一个图片、一个表格等拥有特别的动画效果。如单击鼠标后图片由下逐渐向上移动，直到全部显示在幻灯片中这样的效果。这种效果被称为“自定义动画”。其设置步骤如下。

（1）单击图 15.43 任务区中的“自定义动画”项，打开自定义动画设置页。

（2）在幻灯中单击要设置动画效果的元素元素，如上例中插入的表格，使其选中。

（3）单击“自定义动画”项中“添加”按钮，出现如图 15.47 所示的对话框。

在该对话框中，可做如下设置。

- 在“基本”列表框中单击任一选项，就可以在幻灯片中看到相应的效果。
- 在“强调”选项卡中设置一些特殊的效果，如“透明”、“闪烁”等。
- 在“退出”选项卡中设置退出时该项的动画效果。
- 在“运动路径”标签页中使该项沿着特定的路径运动的效果。

6．幻灯的切换效果

两张幻灯片之间的切换时的特效也是在主界面中任务区“幻灯片切换”中设置。单击“幻灯片切换”，打开如图 15.48 所示设置项。在最上端的列表中选择了适当的切换效果后，可以单击最下面的“应用于所有幻灯片”使所有的幻灯片切换均使用这一特效。



可以将幻灯片文件上传到共享网站 http://www.slideshare.net/ 与他人共享。可以使用命令“sudo apt-get install openoffice.org-ogltrans”安装 3D 切换特效。



图 15.47　自定义动画



图 15.48　幻灯的切换效果

15.5　IBM Lotus Symphony 办公套件

IBM 旗下的 Lotus Symphony 也是一套免费的可运行在 Linux 平台的办公软件。可以用其进行文字处理排版、创建电子表格、制作幻灯片。IBM Lotus Symphony 是基于 Eclipse（IBM 旗下一款强大且免费的编程 IDE）技术的，开放了编程接口，可以开发出符合特定业务需求的插件，而使 Lotus Symphony 更加强大。

对于普通用户关心的是否能打开和保存为 Microsoft Office 格式的文件，答案当然是可以。这种功能使得与其他用户交流也毫无障碍。

作为开源世界的重要成员，IBM Lotus Symphony 采用 OASIS（The Organization for the Advancement of Structured Information Standards，结构化信息标准推动组织）制定的 ODF（Open Document Format，开放文档格式）作为默认文档存储格式。这种格式的文本是通用格式，任何遵循标准协议的软件均可打开。

15.5.1　下载安装

这款软件在各个源里均没有包含，所以需要自行安装。安装的具体步骤如下。

（1）在 IBM 的网站下载该软件的二进制包。下载地址是 http://www-01.ibm.com/software/cn/lotus/symphony/download.html?S_TACT=105AGX52&S_CMP=dwcnl。

（2）下载回来是个名字类似于 IBM_Lotus_Symphone_linux_v1.zip 的压缩文件，要先对其进行解压。

（3）解压后得到两个文件，一个叫 IBM_Lotus_Symphony_linux.bin，是安装的主文件，另一个名为 updateSite_zh_CN.zip，是用来汉化的文件。

（4）将 IBM_Lotus_Symphony_linux.bin 的权限更改为可执行。最简单的做法是在 Shell 中使用命令“sudo chmod +x IBM_Lotus_Symphony_linux.bin”。

（5）因为在安装过程中需要设置 JRE，需要 root 用户权限，所以要使用命令 sudo ./IBM_Lotus_Symphony_linux.bin 来安装。如果直接使用./ IBM_Lotus_Symphony_linux.bin，会提示错误的。

（6）安装过程启动后的界面如图 15.49 所示。单击 I accept both the IBM and the non-IBM terms 同意该协议，以后的选项一般取默认值，一直单击 Next 按钮就可以了。最后完成时，单击“Finish”未完成安装。

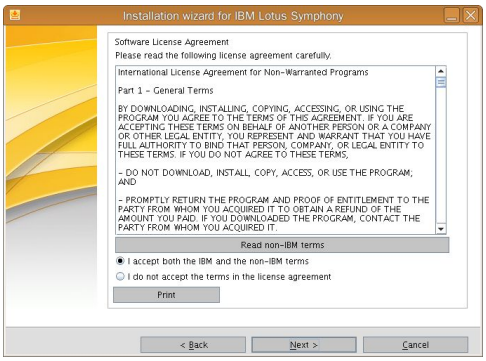


图 15.49　安装 IBM Lotus Symphony

（7）安装好以后，还要修改 lotus 目录的使用权限，使用命令 sudo chown -R 用户名 ～/.lotus/ 修改。



软件的安装位置在一个很复杂的目录中，其位置是/opt/ibm/lotus/Symphony/framework/shared/eclipse/ plugins/com.ibm.productivity.tools.standalone.addin_3.0.1.20080602-1703/apps/，软件名是 IBM Lotus Symphony。

（8）软件的运行主要通过选择“应用程序”|“办公”|“IBM Lotus Symphony”命令来运行，有的用户安装后单击菜单无法运行程序，可以直接在终端中使用命令来运行，命令有点长，不过大多可用自动未完成功能来实现输入，命令是/opt/ibm/lotus/Symphony/framework/ shared/eclipse/ plugins/com.ibm.productivity.tools.standalone.addin_3.0.1.20080602-1703/apps/IBM\ Lotus\ Symphony。

15.5.2　使用简介

IBM 是业界大亨，所以其软件也很有自己的特色。其界面感觉更像是个网络浏览器，如图 15.50 所示。从图 15.50 中看到，该软件的界面与其他办公软件有很大的不同。

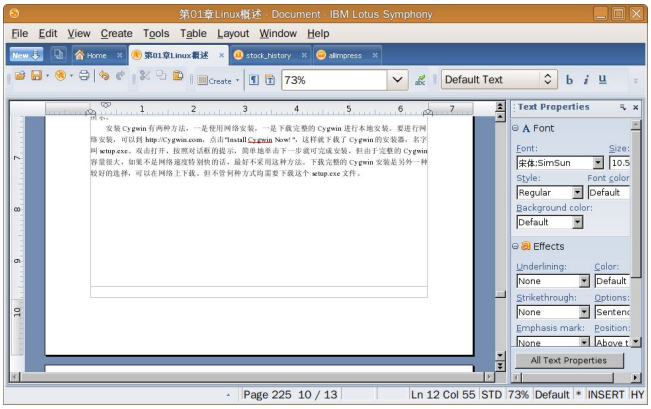


图 15.50　Symphony 字处理界面

Symphony 的特点如下。

- 字体、字号等设置项目均放在界面的右侧。
- 插入图像的功能在图中所指的区域和“Create”菜单中实现。
- “样式”可以设置当前段落使用的样式，如“标题 1”、“标题 2”、“正文”之类。
- 在“显示比例”左边的小按钮功能类似于其他办公软件中的“格式刷”。
- 在“格式刷”左边的按钮控制是否显示控制字符，如回车等。
- 不知为什么，Symphony 将显示比例的位置放得很醒目。
- 页面中菜单下方是标签页，显示了当前打开的所有文件，可以单击页名来切换。
- 显示打开的所有页面的缩略图。可以方便地查看当前打开的所有文件，还提供了搜索功能以文件名查找打开的文件。当前打开的缩略图如图 15.51 所示。可以看到，除了左上的第一个文件没有打开是空白外，打开了所有 Symphony 支持的三种文件，右上是字处理软件，左下是电子表格，而右下则是幻灯片。
- 选择 Tools|Gallery 命令后，会出现插画库，库里的图片很丰富。

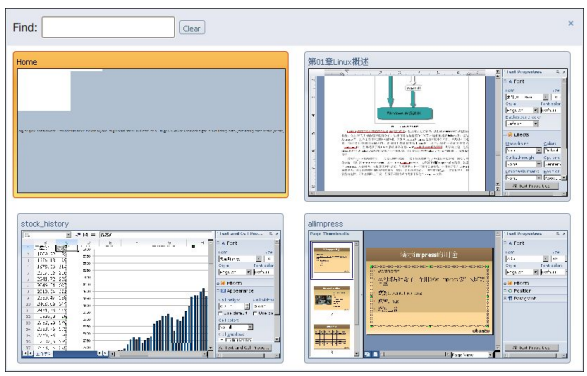


图 15.51　缩略图

熟悉了其界面后，会发现相应的功能其实与其他办公软件很是类似。如果熟悉了其他办公软件，对于 Symphony 也是很容易上手的，而且还有些相对来说很有特色的功能。有 IBM 的背书，相信 Symphony 的性能会很不错。



相对来说，IBM Lotus Symphony 与 Microsoft Office 的比 OOo 稍微高些，但系统稳定性又要比永中 Office 好些，是一款比较优秀的办公软件套件。

15.6　永中 Office 办公套件

永中 Office 在 Linux 办公软件领域也算是一个重量级的软件，由无锡永中科技有限公司出品。其主要特点是界面与 Microsoft Office 的相似性，而且具体的操作也极类似，对 Office 文件兼容性也在各个办公软件中也算是很不错的。比较适用于想在 Linux 中找到 Microsoft Office 感觉的用户。相对来说，永中 Office 有如下特点。

- 与 Microsoft Office 的界面相似程度高，很容易将其使用经验转换到永中 Office。
- 对 Microsoft Office 的文件格式支持度高，打开大多数复杂格式的文件均不会出现格式

错误的问题。

- 通过国家首批认证，完美支持中文标准办公软件格式。
- 系统稳定性有待提高。
- 其保存文件使用 eio 格式，这种格式其他软件无法识别。而且系统最让人无法接受的是如果编辑的文件格式不是 eio，则无法使用“保存”功能直接保存文件，必须使用“另存为”功能保存，这给正常使用带来极大的麻烦。
- 不提供对开放文档格式（ODF）支持，如 ODP、ODS、ODT。而这些文档格式在非微软世界的使用极其广泛，这样就将很大一部分潜在的用户挡在了门外。

截至 2008 年 10 月，其最新版本是 V2009 个人版，实际开发版本号为 5.0.1059.101。个人用户可免费下载试用，试用期是 30 天。不过对 Linux 用户而言，30 天之后还可照常使用，不过在启动时多了一个提示，单击后可正常使用。其具体安装步骤如下。

（1）在如下地址下载 http://218.90.147.70/EverMore/EIOffice2009/EIO2009_Trial_ZH_Lin.tar.gz。

（2）将下载的压缩文件解压。所有的程序文件会放置在以实际开发版本号为名的目录中，如～/software/5.0.1059.101ZH.L1。

（3）进入该目录，改变安装文件 setup.sh 属性为可执行。可在文件管理器中进行，也可在终端中输入 chmod +x setup.sh。

（4）安装需要进行 JRE 的设置，所以需要 root 权限，因此使用命令 sudo ./setup.sh 安装。

（5）安装开始，会出现欢迎界面，单击“下一步”按钮开始安装。一般来说，在其他页面均可接受默认值。

（6）安装完成后，可在命令行输入 eio 或选择“主菜单”|“办公”|“启动永中集成 Office2009”命令运行永中 Office。其界面如图 15.52 所示。

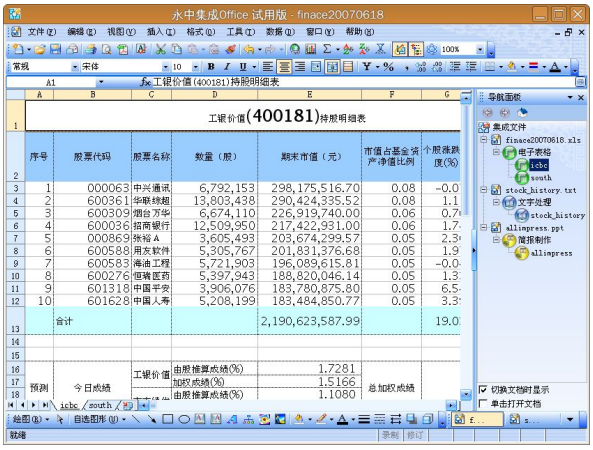


图 15.52　永中 Office 主界面

界面与 Office2003 基本是一样的，熟悉其他办公软件的用户应该可以轻易上手，就不再介绍了。



由于永中 Office 的封闭的特性，如不支持 ODF，也无法简单地将文件保存为 Microsoft Office 的格式，所以并不推荐普通用户使用。

15.7 文书浏览

在实际的工作与生活中，经常要打开许多格式的文档，但是 Ubuntu 中默认对这些文件支持度并不好，如无法正常显示简体中文的 PDF 文件，而且不支持很常见的一种电子书格式 CHM 文件。本节的内容主要是安装 PDF 的简体中文支持，以及安装打开 CHM 的文件。

15.7.1 正确显示中文 PDF 文件

PDF 文件格式是国际通用的文档格式。其特点是显示的稳定性，就是在任何系统上显示的内容均保持一致，大多数科技文献均是以 PDF 格式保存。但由于汉字显示的复杂性和有的 PDF 制作并不符合规范，所以导致了有些中文 PDF 文件打开会成为乱码。

Evience 是 Ubuntu 系统默认打开 PDF 文件的软件。其具有体积小巧、占用系统资源少等优点，但中文显示却是其软肋。不过，只要安装一些补丁，就可以让 Evience 正确显示各种中文 PDF。安装这些补丁的命令如下：

```
sudo apt-get install xpdf-chinese-simplified xpdf-chinese-traditional poppler-data
```

安装后，再打开原来乱码的中文 PDF，就会看到出现了美丽的中文。

15.7.2 打开 CHM 文件

CHM 文件是标准的 Windows 帮助文件格式，现在许多文档，如电子书等均使用了 CHM 格式。打开 CHM 文件有许多方法，一种是安装 Firefox 的 CHM 插件，另一种是安装独立的 CHM 阅读器，不过，大多数的 CHM 阅读器对于中文支持并不好，打开速度也比较慢，所以推荐使用 Firefox 的扩展来打开 CHM 文件。

1．安装阅读 CHM 文件的 Firefox 扩展

Firefox 的 CHM 阅读插件名字叫 CHM Reader。安装该插件的步骤如下。

（1）运行 Firefox，选择 Tools|Add-on 命令。

（2）在 Get Add-on 标签中输入 CHM Reader，搜索结果出现该插件。

（3）单击 Add to Firefox 按钮进行安装，按要求重新启动 Firefox，就会在其 File 菜单项中看到多了一项 Open CHM file 的菜单项。单击该菜单项，会出现打开文件对话框，选择一个 CHM 文件，就可进行浏览了。

不过，在阅读前，应该选择 Firefox 的菜单 View|Sidebar|CHM reader 命令或直接按下 Ctrl+E 快捷键打开 CHM 阅读插件的侧边栏。侧边栏显示了 CHM 文件的索引目录和搜索等功能的 Firefox 如图 16.53 所示。

2．CHM 阅读器

Ubuntu 中的 CHM 阅读器有几款，如 GnoChm、ChmSee、KchmViewer 等，经过实际使用经验，笔者推荐的是 ChmSee。在终端中输入 sudo apt-get install chmsee，按下 Enter 键执行后就可完成安装。安装后打开 CHM 文件如图 16.54 所示。该软件对中文识别能力与 CHM 文件解析能力均很强大，可正确识别大部分 CHM 文件及中文编码。

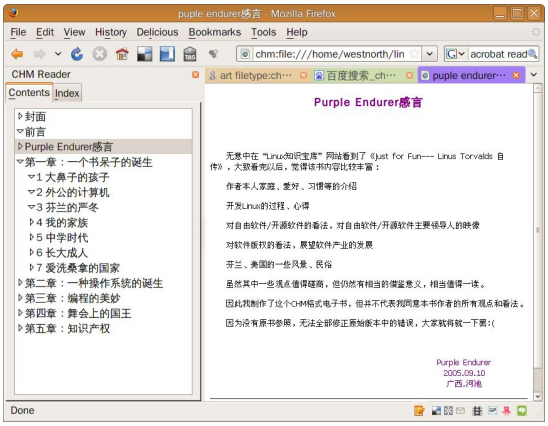


图 16.53　Firefox 的 CHM Reader 插件

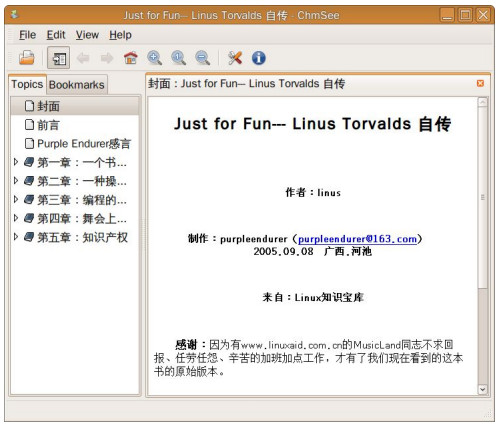


图 16.54　CHMSee 界面



由于 PDF 与 CHM 格式的文本是网络常见的电子书格式，这里也就着重介绍了如何正确打开这两种格式文本的方法。至于其他如 exe 格式的电子书，则应该通过 wine 打开。

15.8　小结

本章介绍了几款比较常见的 Office 软件的使用以打开常见格式电子文档的方法。本章的重点内容是 OOo 办公软件的使用，包括文字处理、电子表格、演示文档。因为这些均是工作中需要经常使用的基本技能。熟练掌握，可以从很大程度上提高工作效率。

第 16 章　专业排版软件 LaTeX

习惯了 Windows 操作的用户，都已经习惯了“所见即所得”的排版工具。一般来说，普通的办公软件均属于该类型。这些软件的特点是，屏幕看到的内容就是实际输出的效果，在编辑过程中随时可以看到版式的变化。但 Linux 中有一款强大的“所想即所得”的排版工具，就是大名鼎鼎的 LaTeX。LaTeX 是由一些带格式的文本组成的，这些文本实际上是一些排版指令，用特定的命令将其转换后才可看到版式效果。但其格式控制极其灵活方便，所以称之为“所想即所得”。本章就来介绍一些 LaTeX 相关的知识。

16.1　LaTeX 的特点

1978 年，美国斯坦福大学教授 Donald E. Knuth，编写了巨著《计算机程序设计艺术》（The Art of Computer Programming）。但在出版时却发现当时粗糙的排版水平使得这部巨著印刷远远称不上是 Art。生气之余，他开始编写高质量的排版软件，就是 TeX。由于这传奇般的诞生经历，使得 TeX 极其关注于排版及印刷质量（是的，这是个排版软件，但同时还管理了打印质量），所以在学术领域，尤其在数学、物理学和计算机科学界得到了广泛的使用。TeX 的特点如下。

- TeX 的一个很有趣的地方在于其版本号。从 TeX 第 3 版开始，之后的升级是在小数点后再加入一个数字，使版本号越来越接近圆周率 π 的值。TeX 目前的版本是 3.141592。
- TeX 是非常稳定的程序。Donald 给任何能在 TeX 程序中发现 Bug 的人给予奖励，现在奖金数额是 327.68 美元。不过，现在看来拿到这 327.68 美元是一个不可能的任务了。
- TeX 是排版数学公式的最优选择，同时也是许多科技期刊、科技出版社的指定格式。如美国数学学会将 TeX 作为向其期刊投稿的优先格式，同时许多一流的出版社如 Addison-Wesley、牛津大学出版社等也给出了 TeX 的模板，向其投递的稿件必须使用 TeX 格式。
- TeX 免费提供源代码，可以自由使用。
- 多平台支持：目前在许多平台上均有 TeX 的实现。
- TeX 的排版输出格式是 DVI（Device Independent）。由名字可以看出，它是与设备无关的，可以转换为 PDF，也可以直接将其打印，同时保证格式完全一致。
- 超强的稳定性：同一个 TeX 文件，由最新版本的程序处理结果与 20 年前的版本处理结果完全一致，这也可以看出其完美的稳定性。
- TeX 内容的可读性：虽然 TeX 的内容实际上是一些程序处理指定，但格式为纯文本，可以用任何的文本编辑软件查看内容。
- TeX 的可编程性，可以自己编写 TeX 命令来实现自己的功能，而这样做其实很简单。
- 灵活性，使用不同的宏包，可以实现许多不同的任务，实现双字节文本的处理（如中、日、韩文字），还可以用来编辑乐谱、棋谱等。

但 TeX 的缺点如下。

- 使用极其复杂，虽然使用熟练后会带来许多乐趣，但是开始接触这样奇怪的系统实在让人有点摸不着头脑。TeX 的命令繁多，学习使用需要大量的时间；而 LaTeX 的学习则要相对简单得多，使用起来也比 TeX 方便快捷。
- 发生错误时，很难理解系统给出的错误信息提示。
- 不是所见即所得，无法方便地在编辑时查看实际的编辑效果。
- TeX 功能虽然强大，排版虽然漂亮，但要从头开始定义一个页面排版样式却不是普通用户所能做到的，一般要专业人员提供特定的模板。
- 不适用于那些图片较多，排版过于复杂的版面，如时尚杂志、艺术画册等。

LaTeX 是基于 TeX 排版引擎的一个排版程序。TeX 提供了 900 多条命令，对于普通使用者来说过于复杂，但 TeX 提供了宏定义功能可以方便地对其进行二次开发。其中比较著名的有美国数学学会推荐的适合于数学家使用的 AMS-TeX 以及适合于一般文章、报告、书籍的 LaTeX 系统。这些系统比 TeX 更为结构化，提供了许多模块化的功能，如“\documentclass{article}”语句声明了一个使用 article 模板的文档，而“\author{westnorth}”指定了文档的作者为 westnorth，这些内容在经过程序处理后均会按照系统指定的方式显示出来。

LaTeX 的标准标志如图 16.1 所示。其中，A 稍为上浮，而 E 略为下沉，在普通文本环境中，使用 LaTeX 来表示。



图 16.1　LaTeX 的标志



由于 LaTeX 的通用性是相对于英语语系的用户而言的。对于东亚语系用户来说，中文等语种的 LaTeX 文件可能无法直接被英文系统的 LaTeX 识别，所以还是存在兼容性方面的问题。这一点，用户也在使用时应加以注意。

16.2　安装 LaTeX 程序

要使用 LaTeX，需先进行安装。LaTeX 中有图形界面的 LaTeX 环境，可以从某种程度上支持“所见即所得”，如 TeXLive 等。但为了更好地理解 LaTeX，我们先安装基本的 LaTeX 程序，然后在命令行中自己编译纯文本格式的 TeX 源文件。

16.2.1　安装核心程序

LaTeX 的核心程序包括 tetex-bin、tetex-base、tetex-extra、cjk-latex，最后一个程序是提供 CJK（China、Japen、Korean 三个国家）文字支持的。安装很简单，在命令行中运行如下命令即可：

```
sudo apt-get install tetex-bin tetex-base tetex-extra cjk-latex
```

这几个包也比较大，总共有 106MB，下载可能需要很长时间，安装也需要时间。但在耐心地等待十几分钟后，一个基本的 LaTeX 环境就建立成功了。

下载安装了这些程序后，我们就拥有一个可以运行的 LaTeX 环境，可以实验所有的 TeX 命令了。我们先以一个简单文件来体验 LaTeX，建立的步骤如下。

（1）打开终端，创建一个用于建立 Latex 的目录。目录名可以任意选取，本例中目录名为 latex，命令是：

```
mkdir latex
```

由于打开终端时，默认的目录是用户主目录，所以这个目录实际上是“～/latex”。

（2）将当前工作目录更改到刚才创建的目录 latex。命令如下：

```
cd latex
```

（3）使用命令 gvim simple.tex 新建一个空白的 TeX 文件，将下面的内容输入到文件中：

```
\documentclass[12pt]{article}
\title{My First file of \LaTeX }
\author{westnorth}
\begin{document}
  \maketitle
  \LaTeX{} base on \TeX{} ,
 new versions is  \LaTeXe.
  \newline
  %这是注释，现在没有安装中文支持，还不能在正文中输入中文
  \begin{eqnarray}
    E &=& mc^2                              \\
    m &=& \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}
  \end{eqnarray}
\end{document}
```

看不懂文件中有些行的意思也没关系，我们将在本节其他小节中介绍这些语句的含义。现在简单地将其输入即可。保存 simple.tex 文件，然后继续在终端中运行下面的命令。

（4）使用 LaTeX 将源 TeX 文件编译为 DVI 格式文件，命令如下：

```
latex simple.tex：使用 latex 命令编译 TeX 文件，并输出与源文件同名的 DVI 文件，本例中 DVI 文件名为 simple.dvi。
```

（5）使用 dvipdfm 将 DVI 文件转换为 PDF 文件，命令如下：

```
dvipdfm simple.dvi：将 DVI 文件转换为 PDF 文件。
```

第 4、5 步也替换为下列一条命令，可以达到同样的效果。

```
pdflatex simple.tex 直接将 tex 转换为 pdf 文件。
```

现在，用任何一款 PDF 查看器打开刚刚的 simple.pdf 文件，即可看到类似图 16.2 所示的效果。

如果是西方语言的用户，到这步已经安装成功。但作为中国用户而言，还有很重要的一个步骤，就是让 LaTeX 支持中文。16.2.2 节就介绍如何安装 LaTeX 的中文支持。

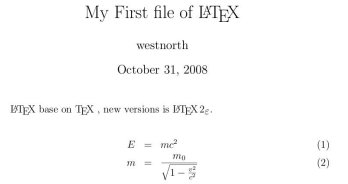


图 16.2　第一个 TeX 文件输出效果

16.2.2　安装中文支持

要使 LaTeX 能够正确识别中文，需要 CJK 包。在 16.2.1 节介绍安装过程时，已经将其安装。但系统现在还没有合适的字体让 CJK 显示中文。本节主要介绍如何创建 CJK 可以识别的

中文字体。创建 CJK 字体可以使用 gbkfonts 软件。软件的下载地址是 http://www.zhyfly.org/projectsz/source /gbkfonts。然后参照如下步骤，即可生成合适的中文字体：

（1）只有用户主目录中的 texmf 目录才是 TeX 可识别的目录，所以必须严格按照下列命令输入来新建目录：

```
mkdir ～/texmf
```

（2）更改当前工作目录到 texmf，使用如下命令：

```
cd ～/texmf
```

（3）执行如下命令将系统中默认安装的文泉驿正黑字体作为 CJK 字体复制过来，并将其重新命名为 zenhei.ttf：

```
cp /usr/share/fonts/truetype/wqy/wqy-zenhei.ttf ./zenhei.ttf
```

这个重命名并不是必须的，这里重命名的原因是为了方便使用。当然，也可以使用其他中文字体，如 simsun.ttf 等。不过，既然使用开源自由的 Linux，就应该使用开源共享的自由字体，而且正黑字体是比较漂亮的。

（4）将 gbkfonts 也复制到当前目录，使用如下命令：

```
cp ～/software/gbkfonts ./
```

这里的～/software/目录是保存 gbkfonts 的目录，读者可以使用自己保存文件的目录来取代该目录。

（5）更改 gbkfonts 的执行权限为所有用户均可运行，命令如下：

```
chmod 755 gbkfonts
```

（6）用 gbkfonts 将 zenhei.ttf 字体转换为 LaTeX 可识别的格式，命令如下：

```
./gbkfonts zenhei.ttf zenhei
```

这一步会在目录中生成包含 CJK 字体信息的许多文件和目录。

（7）使用 root 身份打开 updmap.cfg 文件，命令如下：

```
sudo gedit /var/lib/texmf/web2c/updmap.cfg
```

打开文件后，在文件内容最后面加入“Map cjk.map”，然后保存退出。

（8）刷新 LaTeX 的字体信息，使新安装的字体被 LaTeX 识别，命令如下：

```
sudo texhash; sudo updmap
```

经过几分钟的等待后，CJK 中文字体就成功地添加进了 LaTeX。下面开始创建第一个中文 LaTeX 文件，具体步骤如下。

（1）更改目录到 latex，使用命令：

```
cd ～/latex
```

（2）继续新建一个 TeX 文件，使用命令：

```
gvim zh.tex
```

（3）将下面的示例 TeX 内容输入到 zh.tex 文件中。

```
\documentclass{article}
\usepackage{CJK}
\author{westnorth}
\title{My First file of \LaTeX }
\begin{document}
\begin{CJK}{GBK}{zenhei}
\maketitle
使用文泉驿正黑字体的 \LaTeX 文件。
\newline
%使用镕字是因为其是 GBK 中的学，如果该字可以正常显示，则其他 GBK 汉字显示也无问题
如果看到了镕字，说明可以正常显示 GBK 汉字
\begin{eqnarray}
    E &=& mc^2                                \\
    m &=& \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}
\end{eqnarray}
\end{CJK}
\end{document}
```

（4）由于安装的 CJK 支持只识别 GBK 编码的文件，而 Ubuntu 的文件默认编码是 UTF-8，所以还需要将文件格式轮换一下，才可正确进行编译。在 gvim 中按下 Esc 键，进入命令行模式，输入 set fileencoding=gbk，如图 16.3 所示。

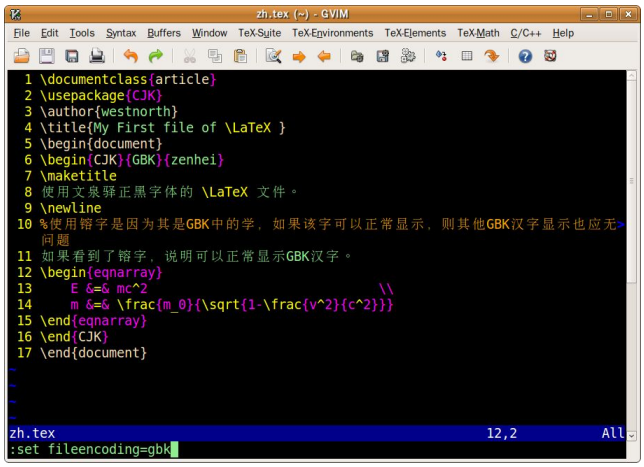


图 16.3　设置文件编码

单击回车运行此命令后，就将文件编码重置为 GBK。现在就做好了编译中文内容 LaTeX 文件的准备。

（5）编译 LaTeX 文件，使用命令如下：

```
pdflatex zh.tex
```

编译成功后，就可以打开与源文件同名的 PDF 文件查看其内容了。也可以在命令行中输入如下命令：

```
evince zh.pdf
```

输出的 PDF 文件应该类似于图 16.4 所示。



图 16.4　中文 LaTeX 的输出效果

图中显示的汉字就是文泉驿正黑字体的效果，应该说还算是比较漂亮的。这样，在自由的 LaTeX 世界里也拥有了一款自由而美观的汉字字体。



使用这种方法编写的 LaTeX 文件只能使用 GBK 编码保存，这也是许多用户出现编译无法通过的主要原因。如果正确安装了 LaTeX 与 CJK 支持，仍然出现 LaTeX 文件无法编译，应该将该 LaTeX 文件保存为 GBK 编码再试着编译，应该可以解决大部分问题。

16.2.3　安装 vim 的 LaTeX 插件

现在安装了 LaTeX 的核心处理系统，也可以让 LaTeX 处理中文字符。但是如果每次都从头开始，用纯粹的文本编辑器编写 LaTeX 效率就很低。如果经常使用的编辑器是 vim，可以安装一个 vim 的 LaTeX 插件，这样可以大幅度提高使用效率。这个插件的名字叫 VIM LaTeX Suite。

1．插件安装

可按如下步骤进行安装。

（1）在如下地址下载 http://vim-latex.sourceforge.net/download/latexSuite20060325.tar.gz。

（2）下载后将其解压到～/.vim 目录。

（3）打开.vimrc，然后将下面的内容添加到文件中：

```
filetype plugin on
set shellslash
set grepprg=grep\ -nH\ $*
filetype indent on
let g:tex_flavor='latex'
```

（4）修改“～/.vim/ftplugin/tex.vim”文件，在其中加入这些内容：

```
" this is mostly a matter of taste. but LaTeX looks good with just a bit
" of indentation.
set sw=2
" TIP: if you write your \label's as \label{fig:something}, then if you
" type in \ref{fig: and press <C-n> you will automatically cycle through
" all the figure labels. Very useful!
set iskeyword+=:
```

（5）打开 vim，然后按下 Esc 键进入命令行模式，输入 helptags ～/.vim/doc 在 vim 中创建该插件的帮助文件。这样，就可以在 vim 中输入 help latex-suite 查看帮助文档。

2．插件功能简介

安装结束，Gvim 看起来并没有什么变化。但只要打开一个 TeX 文件，就会发现 Gvim 多了一些菜单，这是因为 LaTeXSuite 插件自动检测打开文件类型。如果是 TeX 文件，才出现处理 LaTeX 的功能菜单。打开 TeX 后 Gvim 界面如图 16.5 所示。

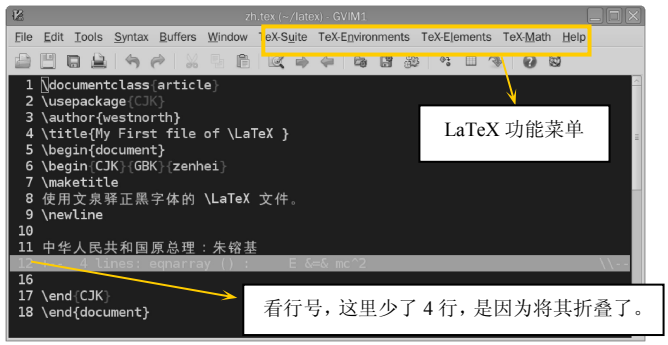


图 16.5　打开 TeX 文件的 Gvim 界面

TeX-Suite 菜单里是主要的功能区，里面的常用菜单项功能如下所示。

- Packages：插件里面内置的包，可以直接插入。
- Templates：插件内置的模板，用户也自建属于自己的包。
- compile：编译 TeX 文件，这是对每个 TeX 文件均要执行的命令，也是编辑 TeX 文件结束后第一个运行的命令，用该命令可以将 TeX 文件编译成 dvi 格式。如果出现错误，该插件可以给出比较详细的提示。
- view：TeX 文件编译后，就可以使用该菜单项预览排版效果。可以按住鼠标放大显示文件中的某部分，如图 16.6 所示。

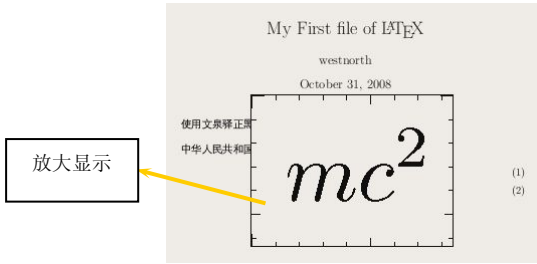


图 16.6　预览 TeX 文件的输出效果

3．自定义模板

TeX-Suite 加入了几个模板。在使用时，直接选择 TeX-suiter|Templates 命令，选择相关的模板就将模板插入当前文档，使用很方便。其默认有四个模板，而且还可自定义模板来方便使用。自定义模板的方法很简单，按如下步骤就可以了。

（1）在目录～/.vim/ftplugin/latex-suite/template/中新建一个名为 article_cn.tex 的文件。该目录是存放模板的地方，任何加入该目录的 TeX 文件均可当作模板使用。

（2）在其中加入如下内容：

```
\documentclass[a4paper]{article}
\usepackage{CJK}
\begin{document}
\begin{CJK}{GBK}{zenhei}
\end{CJK}
\end{document}
```

这是中文环境的模板，注意保存前在 Gvim 中使用命令 set fileencoding=gbk 将文件编码设置为 GBK，这样才可以正确编译。

（3）保存退出后，重启 Gvim，就可以在菜单 TeX-suiter|Templates 中打开该中文模板使用了。

4．宏包管理

如要插入名为 amsmath 的包，选择 TeX-Suite|Packages|Supported|Slimits-chapterbib|Amsmath 命令即可完成插入该包的工作。该菜单列出的宏包均是系统已经安装的包，将宏按字母顺序排列，第三级菜单 Slimits-chapterbib 就标明了“起－始”的包名。

如果包名没有在菜单中出现，应该选择 TeX-Suite|Packages|Supported|UpdatePackage 命令刷新一次，正确安装的包就会出现在菜单中。

插入宏包后，在菜单 TeX-Suite|Packages 次级菜单中会多出两个菜单项，分别是“包名 options”（包含了包的可设置项）和“包名 commands”（包含了宏包的命令集合）。单击菜单项就可插入相关命令，很方便。

5．快捷操作

在使用 TeX-Suite 插入命令后，鼠标会定位于第一个参数中，命令的末尾多了“<++>”的字符，如图 16.7 所示。很多初学者可能会觉得麻烦，其实这正是 TeX-Suite 的特色功能。会出现这种情况：在参数项中输入参数后，按下 Ctrl+J 组合键，就会惊喜地发现鼠标跑出了此命令，而后面跟的“<++>”内容也被删除，其实这个无意义的字符组合定义了鼠标跳出的位置。

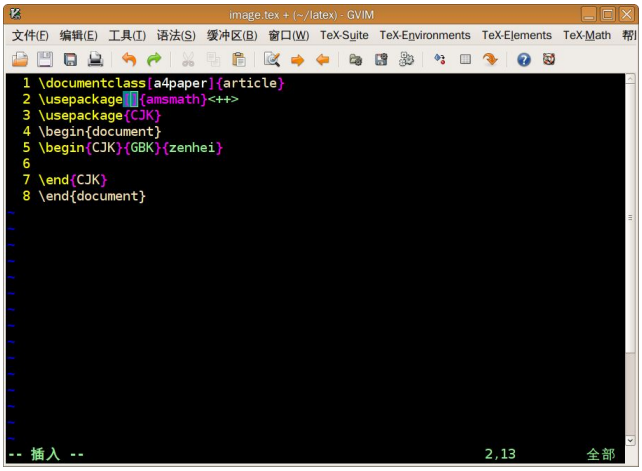


图 16.7　插入命令后的 Gvim

6．调试错误

安装了 TeX-Suite 插件后，如果输入的 TeX 文件格式有错误，就可以在 Gvim 中得到关于

错误格式的提示信息，如图 16.8 所示。错误提示界面分为上、中、下三部分。最上面的部分是 TeX 的源文件，中间的部分是错误的简单提示，单击该错误提示就会跳到源文件中出错的位置。最下面的部分是给出的修正建议，有时还是很有作用的。如图 16.63 所示的建议说“我在前面看见过’$$’，所以我建议你在这儿也输入’$$’。”

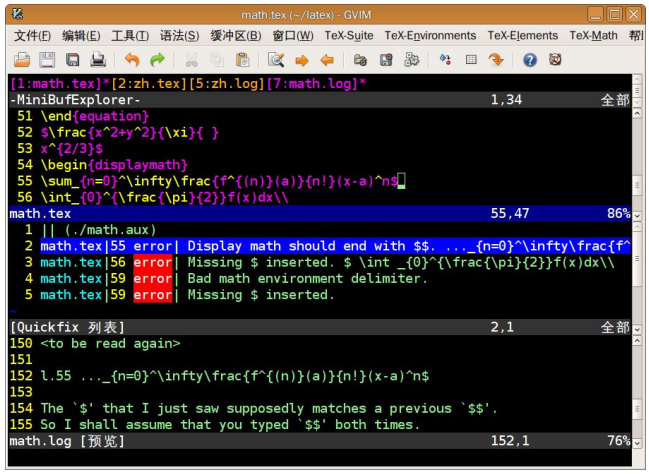


图 16.8　Gvim 中得到的 TeX 文件错误信息

7．插入数学符号

LaTeX 是以输出公式美观而著称的，但同时其公式太多，所以 TeX-Suite 插件提供了简单的插入数学图形的方法。单击菜单“TeX-Math”，可以看到，其中列出了绝大多数的数学符号，可以单击菜单项插入。

16.2.4　安装 Emacs 的 LaTeX 插件

Emacs 作为一款强大的文本编辑器，当然也对 LaTeX 提供了良好的支持。但是 Emacs 内置的 LaTeX 支持功能比较弱，所以还需要安装插件来支持其工作。Emacs 的 LaTeX 插件名叫 auctex。可以在 http://ftp.gnu.org/pub/gnu/auctex/auctex-11.85.tar.gz 下载。这是一个源代码文件，需要进行编译才能使用。下面是具体的安装步骤。

（1）将下载的压缩文件包解压后，得到一个 auctex-11.85 的目录。

（2）在终端进入该目录，运行./configure -with-emacs;make;sudo make install 就可以完成该插件的安装。

（3）安装后，还需要对 Emacs 进行设置。打开～/.emacs，在其中加入如下代码：

```
(load "auctex.el" nil t t)
(load "preview-latex.el" nil t t)
(setq TeX-auto-save t)
(setq TeX-parse-self t)
(setq-default TeX-master nil)
(add-hook 'LaTeX-mode-hook 'LaTeX-install-toolbar)
(add-hook 'LaTeX-mode-hook 'turn-on-auto-fill)
(add-hook 'LaTeX-mode-hook 'turn-on-reftex)
```

这样，在用 Emacs 打开 TeX 文件后，就可以打开插件的菜单。而且 Emacs 有一项很有特色的功能——预览功能，可以实时预览正在编辑的数学公式和图片文件，如图 16.9 所示。

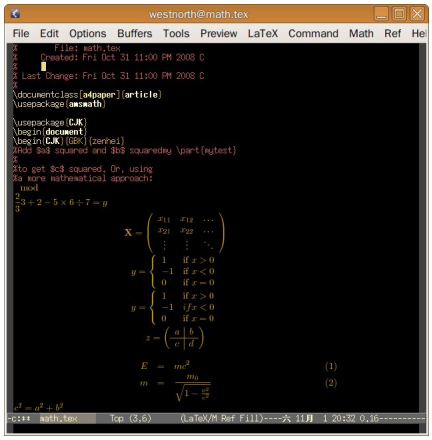


图 16.9　Emacs 中的预览功能



Emacs 的 LaTeX 插件虽然提供了文件预览的功能，但笔者认为其在包管理等方面的易用程度还不如 vim 的插件，读者可以根据自己的喜好选择适当的 LaTeX 编译环境。

16.3　基础知识

前面通过简单的例子看到了如何建立 LaTeX 文件，并可将其输出为漂亮的 PDF 文件。但对于 LaTeX 文件的格式与命令的内容，还是看不懂，故本节将介绍 LaTeX 的基本知识。

LaTeX 文件实际上是一组程序处理语言，其有着严格的格式定义。同时，LaTeX 文件又是纯文本文件，可用任何文本处理程序打开其所有内容。因此，其既是纯文本文件，在其中又包含着程序处理语言，必须要有一种方法来区别其命令与文本。

一般来说，LaTeX 的命令是以“\”开始的一些字符串，格式是“\命令[可选参数] {必选参数}”，如“\documentclass{article}”就是一条命令。没有以“\”开始的内容就是实际会显示的文本。下面以一个简单的 TeX 文件来介绍 TeX 文件的结构和一些基本的命令。

16.3.1　TeX 基本文件结构

如下所示是一个最基本而且比较简单的 TeX 文件。不过其虽然简单，却基本包含了 TeX 文件的基本要素，具体要素将在下文予以介绍。文件内容如下：

```
\documentclass[a4paper]{article}
\usepackage{CJK}
\begin{CJK}{GBK}{zenhei}
\begin{document}
你好
\end{CJK}
```

```
\end{document}
```

下面按照其基本结构对文件进行解析。

1．文件头

LaTeX 创建文件时，须首先知道要创建的文档的类型，所以在文件的第一行须指定文档类型。指定文件类型使用类似于这样的命令“\documentclass[a4paper]{article}”。这个命令是所有 TeX 文件开头必须声明的，用来通知 TeX 程序该文档的类型和页面设置等基本情况。我们将这条命令称为文件头。本例中文件头指定的文档类型是 article，纸张类型是 A4。这些内容均有许多可选项。LaTeX 可用的文档类型（必选参数）如下所示。

- article：用来排版论文，普通的文章、报告等。
- report：用来排版比较长的报告、比较短的书稿、及论文等。
- book：用来排版书籍。
- slides：用来排版幻灯片。

而 documentclass 的可选参数如下。

- 10pt、11pt、12pt：文档的字体大小，默认是 10 磅大小的字。
- a4paper、letterpaper、a5paper、b5paper、executivepaper 和 legalpaper：设置纸张的大小，默认为 letterpaper。
- titlepage、notitlepage：指定是否在文档标题后开始新的一页。
- onecolumn、twocolumn：指定文档以单列或双列方式排版文档。
- twoside、oneside：指定文档为双面或单面格式。
- openright、openany：指定新章的开始方式，openright 是仅在右边页开始，而 openany 是指不限制新章的开始位置。

2．文件首部

文件首部用来完成初始的设置工作。文件首部是指文件头后，\begin{document}命令前的部分。本例中文件首部有两条命令，“%”号后为注释。

```
\usepackage{CJK}          %使用宏包 CJK。
\begin{CJK}{GBK}{zenhei}'   %开始 CJK 中文环境，并默认使用 zenhei 字体。
```

文件首部一般包括引用宏包的命令，标题的设置的命令，作者姓名的设置命令等。

3．正文

正文是从\begin{document}开始到\end{document}结束的部分。这是 TeX 文件的主体，文字、表格、图片、公式均包含在正文中。

16.3.2 关于 TeX 文件的预备知识

LaTeX 包含大量的命令，大多数命令均可在使用中逐渐掌握。本小节将介绍一些在使用前应该有所了解的基本设置命令。

1．环境

如果一些代码需要一个共同的环境来运行，就可以将这些代码都包含在一个环境中。环境命令的使用格式是“\begin {环境名} …… \end {环境名}”。环境中的代码具有某些共性，如 itemize 环境中以\item 开头的文本会加上列表符号。

2．排版前的准备知识

在 LaTeX 中排版不是一件很轻松的工作，而且其在排版时有些设置也是很独特的，下面是一些关于排版方面的知识。

- LaTeX 忽略代码中的连续空格和制表符，将其作为一个看待。而且 LaTeX 的命令会将其后的空格消灭掉，所以要在 LaTeX 命令后使用空格须使用“{ }”。
- LaTeX 用空行分段，连续的空行，将其作为一个看待。
- #、$、%、^、&、_、{、}、～、\：这几个是 LaTeX 中的关键字，不能直接在文本中使用。使用时必须在其前加上“\”，如“ \$ ”。
- LaTeX 另起一行须强制使用“\\”命令。用“*”是断行并禁止分页，“\newpage”开新页。
- 左引号是键盘左上角的“`”键（数字 1 左边的键），右引号是普通的引号“’”，双引号是将其用两次。

3．宏包的安装

为了增强 LaTeX 的功能，开发者们开发出了许多完成特定功能的宏包。通过宏包的使用，可以达到事半功倍的目的。系统中安装有一些宏，但很多时候还需要使用其他外部宏。宏在使用前，需进行一定的设置，以让系统识别。安装宏的步骤如下。

（1）检查宏包的结构与文件，如果有一个后缀为.ins 的文件，就接着进行第 2 步，如果里面有后缀为 sty 的文件，则直接进行第 3 步。

（2）在终端进入到下载的宏包目录，运行命令 latex *.ins，这样就会生成后缀为 sty 的文件。

（3）把后缀为.sty 的文件移到 LaTeX 能够识别的目录，如“～/texmf /tex/LaTeX”目录。

（4）在终端中运行 texhash 命令，让 LaTeX 更新其宏包接口，使得新安装的寡欢包可以正常使用。



LaTeX 宏包主要的下载网站是 http://www.ctan.org/。

16.4　LaTeX 的文档结构

如果使用了 article 风格，也就是说，在\documentclass 中指定文档类型为 article，那么可在文档中使用以下的分节命令。

- \section{...}：文章的节。
- \subsection{...}：文章的次级节。
- \subsubsection{...}：文章的第三级节，这级的节不编号。
- \paragraph{...}：文章的段落。
- \subparagraph{...}：文章的次级段落。

如果文档选择了 report 和 book 风格的话，还有另外两个分节命令。

- \part{...}：文章的篇，如本书的“第一篇基础篇”。
- \chapter{...}：文章的章，如本书的“第 1 章 Linux 概述”。

这些分节命令会自动进行编号，所以对于修改是很方便的，因为就不会因为插入一个图表而对其他的图表重新编号了。如使用下面的源文件，编译出的效果类似于图 16.10 所示。

```
\documentclass{book}
```

```
%使用了 book 样式，字体没有设置，取默认值 10pt.
\usepackage{CJK}
\author{westnorth}
\title{Linux from Beging to Master }
\begin{document}
\begin{CJK}{GBK}{zenhei}
\maketitle
\part{办公软件篇}
%使用 part 命令，该篇名为“办公软件篇”
\chapter{排版软件}
%使用章节命令
    \section{专业排版工具 LaTeX}
%定义节
    TeX：1978 年，斯坦福大学教授 Donald E. Knuth，编写了巨著《计算机程序设计艺术》(The Art of Computer Programming)，但在出版时却发现当时粗糙的排版水平使得这部巨著印刷远远称不上是 Art。生气之余，开始编写高质量的排版软件，就是 TeX。
    \subsection{LaTeX 的文档结构}
%使用次级节
    \paragraph{
%使用段落
LaTeX 文件实际上是一组程序处理语言，所以其有着严格的格式定义。同时，LaTeX 又是纯文本文件，可用任何文本处理程序打开其所有内容。但既然其是纯文本，而又是程序处理语言，所以必须要有一种方法来区别其命令与文本。
}
\subsubsection{math 公式输入}
%这三级节，不参与编号
\section{IBM Lotus Symphony}
IBM 旗下的 Lotus Symphony 也是一套免费可运行在 Linux 平台的办公软件。可以用其进行文字处理排版，创建电子表格、制作幻灯片。IBM Lotus Symphony 是基于 eclipse(IBM 旗下一款强大且免费的编程 IDE)的，开放了编程接口，可以开发出符合特定业务需求的插件，而使 Lotus Symphony 更加强大。
\subsection{下载安装}
    \paragraph{
这款软件在各个源里均没有包含，所以需要去 IBM 的网站自行下载，下载地址：http://www-01.ibm.com/software/cn/lotus
}
\begin{eqnarray}
   E &=& mc^2                              \\
   m &=& \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}
\end{eqnarray}
\end{CJK}
\end{document}
```

用 pdflatex 命令将其直接编译为 PDF 格式，其 PDF 文件打开显示应该类似图 16.10。

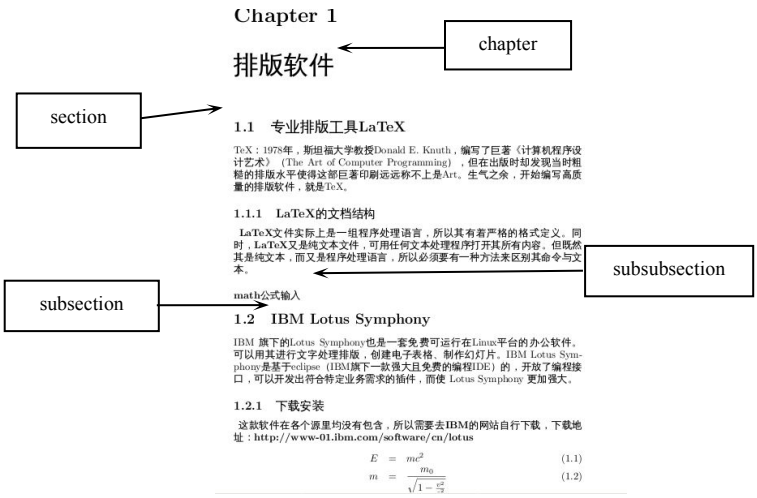


图 16.10　带章节自动编号的文件输出效果

16.5　LaTeX 的公式排版

前面介绍过，LaTeX 被誉为最佳公式排版工具。所以知道如何正确地输入数学公式是很重要的学习内容。图 16.11 是一个公式输出的示例。

1．输入格式

数学公式被作为一个命令输入，经过处理后输出为美观的公式。所以其要遵守严格的格式限定。段落中的数学公式应该在以下任意三组符号之间。

- \（和\)。如“\（a\)”，没错，单个字母也可作为公式的一部分输入。
- 两个$之间，如“$c^{2}=a^{2}+b^{2}$”。
- \begin{math}和\end{math}。
- \[和\]。
- \begin{displaymath}和\end{displaymath}。这个在显示比较复杂的公式时经常使用，会以一种更符合数学公式显示格式的方法来显示公式，如 16.11 所示。

这是使用$作为定界符时输出的内容
$\lim_{n\to\infty}\sum_{k=1}^{n}\frac{1}{k^3}=\frac{\pi^2}{5}$
这是使用'displaymath'作为定界符时输出的内容

$$\lim_{n\to\infty}\sum_{k=1}^{n}\frac{1}{k^3}=\frac{\pi^2}{5}$$

图 16.11　公式输出效果

2．公式编号

前面所述的方法输入的公式是没有编号的，如果要给公式进行编号，需要将公式包含在 eqnarray 中。如：

```
\begin{eqnarray}
   E &=& mc^2                                \\
   m &=& \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}
\end{eqnarray}
```

其输出效果如图 16.12 所示：

$$E = mc^2 \tag{1.1}$$
$$m = \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}} \tag{1.2}$$

图 16.12　公式编号效果

但这样输出的编号是没办法引用的，如果要对公式编号进行引用，就需要在给公式编号时使用另外一条命令。然后公式加上标签，通过标签就可以引用该公式的编号。命令类似于这样：

```
\begin{equation} \label{eq:eps}
```

看一个实际的例子，在 TeX 文件中使用下面的语句，得到的输出应该类似于图 16.13。

```
\begin{equation} \label{eq:eps}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^3}
= \frac{\pi^2}{5}
\epsilon > 0
\end{equation}
通过公式 (\ref{eq:eps})我们可以看出
```

$$\lim_{n\to\infty}\sum_{k=1}^{n}\frac{1}{k^3}=\frac{\pi^2}{5}\epsilon>0 \tag{2}$$

通过公式 (2)我们可以看出

图 16.13　在正文中引用公式编号

3．各种数学符号的输入

数学中有许多的特殊符号，如希腊字母、指数与下标、方根符号等。在 LaTeX 中插入这些符号也比较简单。

希腊字母在数学公式中使用得极其广泛，而 LaTeX 当然也可以输出这些漂亮的字符，使用如下命令得到的输出是 $\lambda, \xi, \pi, \mu, \Phi, \Omega$ 。

```
$\lambda，\xi，\pi，\mu，\Phi，\Omega$
```

要在 LaTeX 中使用指数和下标，可以使用符号“^”“_”，例如在 TeX 文件中使用命令：“$x^2+m_2=3x$”得到的输出是 $x^2+m_2=3x$。

使用命令“\sqrt[n]{formula}”来获得方根符号，n 代表方根的幂次，而“formula”代表表达式。该命令可以嵌套使用。如使用如下代码，得到的效果如图 16.14 所示。

```
$\sqrt[3]{x+y}$\\
$\sqrt{ab}$\\
$\sqrt[3]{\sqrt[3]{\sqrt{y}}}$
```

$$\sqrt[3]{x+y}$$
$$\sqrt{ab}$$
$$\sqrt[3]{\sqrt[3]{\sqrt{y}}}$$

图 16.14　方根符号的输出

简短的符号可直接用“/”来表示，而长一点复杂的分数应该使用“\frac{...}{...}”来排版。如使用如下命令可得到的输出是：$\frac{x^2+y^2}{\xi}x^{2/3}$。

```
$\frac{x^2+y^2}{\xi}{ }
x^{2/3}$
```

积分运算符（integral operator）用“\int”来获得。用“\sum”获得求和运算符（sumoperator）。乘积运算符（product operator）可由“\prod”来生成。各符号的上限和下限用“^”和“_”来生成。如以下的代码生成的高等数学中的几个运算符号如图 16.15 所示。

```
\begin{displaymath}
\sum_{n=0}^\infty\frac{f^{(n)}(a)}{n!}(x-a)^n
\int_{0}^{\frac{\pi}{2}}f(x)dx
\lim_{h\to 0}\frac{f(x+h)-f(x)}{h}
\prod_\epsilon
\end{displaymath}
```

$$\sum_{n=0}^{\infty}\frac{f^{(n)}(a)}{n!}(x-a)^n\int_{0}^{\frac{\pi}{2}}f(x)dx\lim_{h\to 0}\frac{f(x+h)-f(x)}{h}\prod_{\epsilon}$$

图 16.15　高等数学中几个运算符号

可以在大多数数学符号前使用四级的 big、Big、bigg、Bigg 前缀来改变符号的大小。如下代码的符号的放大效果如图 16.16 所示。

```
$\bigg(3 \Big(x/(1-y)\Big)^\pi \bigg) ^{a+x}$\\
$\big(\Big(\bigg(\Bigg(\Bigg)\bigg)\Big)\big)$\quad\\
$\big\{\Big\{\bigg\{\Bigg\{\Bigg\}\bigg\}\Big\}\big\}$\quad\\
$\big\|\Big\|\bigg\|\Bigg\|\Bigg\|\bigg\|\Big\|\big\|$\\
$\big[\Big[\bigg[\Bigg[\Bigg]\bigg]\Big]\big]$
```

$$\Big(3\Big(x/(1-y)\Big)^{\pi}\Big)^{a+x}$$

$$\big(\Big(\bigg(\Bigg(\Bigg)\bigg)\Big)\big)\quad\big\{\Big\{\bigg\{\Bigg\{\Bigg\}\bigg\}\Big\}\big\}$$

$$\big\|\Big\|\bigg\|\Bigg\|\Bigg\|\bigg\|\Big\|\big\|\quad\big[\Big[\bigg[\Bigg[\Bigg]\bigg]\Big]\big]$$

图 16.16　符号的放大效果

```
没有使用分组符号的效果：$x^(3+a)=y$\\
使用了分组符号的效果：$x^{(3+a)}=y$
```

没有使用分组符号的效果：$x^(3+a)=y$

使用了分组符号的效果：$x^{(3+a)}=y$

图 16.17　分组效果

4．数组效果排版

数组效果包括真正的数组，就是括号里数的排列，还包括分段函数的显示等。数组效果排列使用 array 命令。

```
%数组排版可以使用如下代码：
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
\begin{displaymath}
%分段函数排版可以使用如下代码：
y = \left\{ \begin{array}{ll}
 1 & \textrm{if $x>0$}\\
 -1 & \textrm{if $x<0$}\\
 0 & \textrm{if $x=0$}
  \end{array} \right.
\end{displaymath}
\begin{displaymath}
%以下代码在数组内画线
z=\left(\begin{array}{c|c}
 a & b \\
\hline
c & d
\end{array}\right)
\end{displaymath}
```

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots \\ x_{21} & x_{22} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$ ← 数组效果输出

$$y = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \end{cases}$$ ← 分段函数效果

$$z = \left(\begin{array}{c|c} a & b \\ \hline c & d \end{array}\right)$$ ← 在数组内画线效果

图 16.18　数组的输出效果

5．一些常用函数及运算符

一些常用的函数有其预设的格式，可以用下列命令来显示。这些函数包括但不限于\arccos、\cos、\csc、\exp、cosh、\lg、\ln、\arctan、\cot、\log、\sin。

“＋、－、＊、/”四则运算符，可以在 LaTeX 中得到符合日常习惯的符号输出。“＋、－”号可用键盘中相应的键输入，而“＊、 / ”号用“/times、/div”来表示。下面的代码输出效果是：$3+2-5\times6\div7=y$。

```
$3+2-5\times6\div7=y$
```

6．在公式中输入文字

公式是由严格的逻辑定义组成的，其中的文字均被解析为表达式，所以不能像普通文本那

样使用字体和空格等格式。要在公式中输入文字，就须用“\textrm”指令。如以下代码产生如图 16.19 所示的效果，可仔细体会其区别。

```
\begin{displaymath}
y = \left\{ \begin{array}{ll}
 1 & \textrm{if $x>0$}\\
 -1 & if x<0\\
 0 & \textrm{if $x=0$}
  \end{array} \right.
\end{displaymath}
```

7．公式符号列表

数学公式中的符号有数百个，很难记忆。所以我们制作了三个符号对照表，以便查阅。由于符号牵涉显示的问题，所以采用了图形的方式来表示。图 16.20 是希腊字母表，每一列的第一个符号是 LaTeX 文件的输出效果，而后面以“\”开始的字符串是用于 LaTeX 源文件中的命令。

$$y = \begin{cases} 1 & \text{if } x > 0 \\ -1 & if x < 0 \\ 0 & \text{if } x = 0 \end{cases}$$

图 16.19　在公式中正确输入文本

α	\alpha	β	\beta	γ	\gamma	δ	\delta	ϵ	\epsilon
ζ	\zeta	η	\eta	θ	\theta	ι	\iota	κ	\kappa
λ	\lambda	μ	\mu	ν	\nu	ξ	\xi	o	o
π	\pi	ρ	\rho	σ	\sigma	τ	\tau	υ	\upsilon
ϕ	\phi	χ	\chi	ψ	\psi	ω	\omega	Γ	\Gamma
Δ	\Delta	Θ	\Theta	Λ	\Lambda	Ξ	\Xi	Π	\Pi
Σ	\Sigma	Υ	\Upsilon	Φ	\Phi	Ψ	\Psi	Ω	\Omega

图 16.20　希腊字母

图 16.21 是箭头汇总表，每一列的第一个符号是 LaTeX 文件的输出效果，而后面以“\”开始的字符串是用于 LaTeX 源文件中的命令。

图 16.22 是二元关系符与运算符汇总表，每一列的第一个符号是 LaTeX 文件的输出效果，而后面以“\”开始的字符串是用于 LaTeX 源文件中的命令。

$\leftarrow$	\leftarrow	$\Leftarrow$	\Leftarrow
$\Rightarrow$	\Rightarrow	$\leftrightarrow$	\leftrightarrow
$\mapsto$	\mapsto	$\hookleftarrow$	\hookleftarrow
$\leftharpoondown$	\leftharpoondown	$\rightleftharpoons$	\rightleftharpoons
$\Longleftarrow$	\Longleftarrow	$\longrightarrow$	\longrightarrow
$\longleftrightarrow$	\longleftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow
$\hookrightarrow$	\hookrightarrow	$\rightharpoonup$	\rightharpoonup
$\uparrow$	\uparrow	$\Uparrow$	\Uparrow
$\Downarrow$	\Downarrow	$\updownarrow$	\updownarrow
$\searrow$	\searrow	$\swarrow$	\swarrow
$\rightarrow$	\rightarrow	$\longmapsto$	\longmapsto
$\Leftrightarrow$	\Leftrightarrow	$\rightharpoondown$	\rightharpoondown
$\leftharpoonup$	\leftharpoonup	$\downarrow$	\downarrow
$\longleftarrow$	\longleftarrow	$\nearrow$	\nearrow
$\Longrightarrow$	\Longrightarrow	$\nwarrow$	\nwarrow

图 16.21　箭头符号

$\succeq$	\succeq	$\gg$	\gg	$\supset$	\supset		
$\supseteq$	\supseteq	$\sqsupseteq$	\sqsupseteq	$\ni$	\ni		
$\dashv$	\dashv	$\equiv$	\equiv	$\sim$	\sim		
$\simeq$	\simeq	$\asymp$	\asymp	$\approx$	\approx		
$\cong$	\cong	$\neq$	\neq	$\doteq$	\doteq		
$\odot$	\odot	$\bigodot$	\bigodot	$\bigcirc$	\bigcirc		
$\amalg$	\amalg	$\leq$	\leq	$\prec$	\prec		
$\preceq$	\preceq	$\ll$	\ll	$\subset$	\subset		
$\subseteq$	\subseteq	$\sqsubseteq$	\sqsubseteq	$\in$	\in		
$\vdash$	\vdash	$\geq$	\geq	$\succ$	\succ		
∇	\nabla	$\surd$	\surd	$\top$	\top		
$\bot$	\bot	$\|$	\|	$\angle$	\angle		
$\forall$	\forall	$\exists$	\exists	$\neg$	\neg		
$\flat$	\flat	$\natural$	\natural	$\sharp$	\sharp		
$\backslash$	\backslash	∂	\partial	∞	\infty		
$\triangle$	\triangle	$\sum$	\sum	$\prod$	\prod		
$\coprod$	\coprod	$\int$	\int	$\oint$	\oint		
$\ldots$	\ldots	$\cdots$	\cdots	$\vdots$	\vdots		
$\ddots$	\ddots	$\pm$	\pm	$\mp$	\mp		
$\times$	\times	$\div$	\div	$\ast$	\ast		
$\star$	\star	$\circ$	\circ	$\bullet$	\bullet		
$\cdot$	\cdot	$\cap$	\cap	$\bigcap$	\bigcap		
$\cup$	\cup	$\bigcup$	\bigcup	$\uplus$	\uplus		

图 16.22　二元关系符与运算符

看了前面的三个图表，可能许多读者还是不大清楚如何在 LaTeX 文件中使用这些符号。下面使用一个简单的示例来说明这些符号表的使用，通过理解该示例，就可以熟练使用这些符号

来方便科学论文的输入工作了。示例如下所示。

（1）新建一个 TeX 文件，假设名为 zh.tex。

（2）在该文件中加入如下内容：

```
\documentclass{article}
\usepackage{CJK}
\author{westnorth}
\title{Char list}
\begin{document}
\begin{CJK}{GBK}{zenhei}
\maketitle
这是测试字母表的示例文件\\
希腊字母：$\alpha \beta\gamma$\\
箭头：$\leftarrow\Leftarrow\Rightarrow$\\
运算符：$\sum\prod\bigodot$
\end{CJK}
\end{document}
```

（3）参照本章的相关内容将其编译为 PDF 文件。如图 16.23 所示就是该 PDF 文件内容。



图 16.23　符号表测试 PDF 文件

16.6　在 LaTeX 中插入图片

要完成一部美观的文章的排版，图片是不可或缺的。虽然在 LaTeX 中插入图片比较复杂，但与其他排版软件一样，还是可以做出复杂排版效果的。

先来看一个最简单的插入图片的例子。在插入图片之前，要先准备一张图片。其实 png 等图片格式也可。不过，LaTeX 最早支持的图片格式是 eps，而许多科学软件均可以生成 eps 格式的图片文件，因此本例以 eps 图片 cross.eps 为例进行介绍。将图形文件（设文件名为 cross.eps）放置于“～/latex”目录中，然后进行如下步骤。

（1）在 Gvim 中新建一个文件，方法是在终端中运行“gvim ～/latex/image.tex”。由于这是一个以 tex 为后缀的文件，所以 Gvim 会自动打开 TeX-Suite 插件菜单。以下几步操作均使用了 TeX-Suite 插件。

（2）单击菜单“TeX-suiter”|“Templates”菜单项，选择“article_cn”模板，生成基本的中文 TeX 文件。

（3）将鼠标定位于文件中“\documentclass[a4paper]{article}”行之后，单击菜单“TeX-Suite”|“Packages”|“Supported”|“Cite-Graphpap”|“Graphicx”命令，引用 graphicx 包。

（4）按下 Ctrl+J 组合键，跳出包引用的命令。

（5）将光标置于“\begin{CJK}{GBK}{zenhei}”之后，选择菜单“TeX-Suite”|“Packages”|“graphicx command”|“includegraphics”|“\includegraphics[]{<++>}<++>”命令，将该命令插入。

（6）现在光标应该停留在命令的第一个参数中。输入 height=200px，然后按 Ctrl+J 组合键进入第二个参数项，再在其中输入 cross.eps。再按下 Ctrl+J 组合键完成该命令。

现在，文件的内容应该类似于这样，如果使用的不是 Gvim 的插件，也可以自行将所有的内容输入。

```
\documentclass[a4paper]{article}
\usepackage[]{graphicx}
\usepackage{CJK}
\begin{document}
\begin{CJK}{GBK}{zenhei}
\includegraphics[height=200px]{cross.eps}
\end{CJK}
\end{document}
```

（7）由于 pdflatex 命令不支持 eps 格式的文件，所以需要直接使用 latex 来进行编译，然后再将 DVI 文件转换为 PDF 格式输出。在终端中输入如下命令：

```
latex image.tex
#该命令从 TeX 文件生成 DVI 文件，DVI 文件预览时可能看不到图片，因为 EPS 图形并没有真正被加到 DVI 文件中，将其转换为 PDF 文件后就可以看到图片了。
dvipdfm image.dvi
#将 DVI 文件转换为 PDF 格式，现在就可以看到插入图片的文件了。
```

刚才介绍的只是插入图片的最简单的方法，实际上 includegraphics 还有许多很实用的选项，如对图片进行旋转等。使用这个命令“\includegraphics[height=200px，angle=30]{cross.eps}”，就将图片顺时针旋转 30°显示。图 16.24 显示了旋转后的图片与图片的对比。

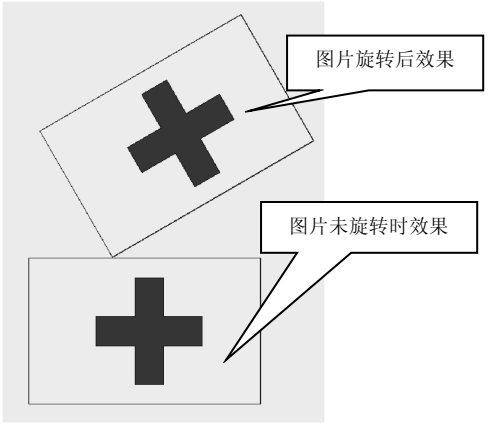


图 16.24　图片旋转后与未旋转对比

\includegraphics 命令的参数还有一些，将其列于下，灵活地使用可以达到更好的显示效果。

- height：指定图像显示的高度。
- width：指定图像显示的宽度。
- angle：指定图像旋转的角度，旋转方向是顺时针，单位是度数。
- bb：设定 BoundingBox 的大小，是图片左上和右下的坐标值，如（10,10,200,200）代表

图片左上角是（10,10），右下角是（100,100）。在使用非 eps 图像时须使用该参数指定图像显示区域大小。

- keepaspectratio：在缩放图像时保持原图像的高度比例。
- scale：指定图像显示的缩放比例，2 代表放大 2 倍，－2 表示缩小 2 倍。
- clip：则不显示视图之外的图像，noclip 则显示视图之外的图像。
- trim：剪切图像，四个数字分别代表从图像的左、上、右、下剪切的尺寸。

16.7 小结

本章介绍的是专业排版工具 LaTeX 的基础知识。对于科研人员而言，LaTeX 可能是必备的专业技能之一。但该排版软件与普通办公软件又有太多的差异，所以使得许多用户从心理上排斥其使用。现在许多院校本科及研究生论文均要求使用 LaTeX 排版，所以下载一个高校的论文 LaTeX 模板，然后练习使用，对掌握 LaTeX 的使用会有很大的益处。

第 四 篇

Linux 开发篇

- 第 17 章　Linux 环境中 C/C++编程简介
- 第 18 章　Shell 编程
- 第 19 章　搭建 LAMP 服务器
- 第 20 章　FTP 服务器的搭建

第 17 章　Linux 环境中 C/C++编程简介

Linux 发源于一群技术高手，所以 Linux 系统有两个特点：一是用户可对系统底层进行设置，另一个就是其与各种编程语言的整合度很高，用户可以轻易搭建一个完美的开发环境。系统自带了 C/C++编译器，所以本章以 C/C++为例介绍了搭建开发环境，并进行简单开发的过程。除此之外，Linux 对 Python 的支持度也很好，可以直接进行 Python 的编程工作。本章不是 C/C++入门，所以并未介绍 C/C++的语法知识，有需要的读者可以参阅相关的专门著作。

17.1　GCC 简介

对于一个高级语言而言，要使代码成为可执行程序，必须要有一个编译器对源代码进行编译。Linux 世界中最为著名的 C/C++编译器无疑就是 GCC 了，其功能强大，支持多种平台，编译出的程序性能优良，是 Linux 每个 C/C++程序员必备的工具。本节介绍一些 GCC 的初级应用程序，掌握本节所述内容，应该具备初步的 GCC 编译能力。

17.1.1　GCC 功能简介

GCC 原始名称为（GNU C Compiler，GNU C 编译器），是 GNU 软件库的基础，也是 GNU 影响最大的软件，同时也是 GNU 的核心部分。本节将介绍 GCC 相关的一些功能。

1．GCC 特色

回顾一下 Linux 的历史就可以知道，Linux 第一个内核就是使用 GCC 编译的。这也符合一般系统开发的规律：提供一个最基本的系统（Linux 内核），再提供一个扩展其功能的工具（GCC），借助于 GCC 编译的各种软件来丰富系统的功能。GCC 在开发出的初期，只能编译 C 语言。但随着 GCC 功能的不断扩充，现在其已经可以编译 C++、Fortran、Pascal、Objective-C、Java，以及 Ada 等许多种类的语言了。所以，GNU 将其名称更改为 GCC（GNU Compiler Collection“GNU 编译器集合”）。本章主要介绍其编译 C 及 C++的方法。

GCC 是遵从 GPL 及 LGPL 许可证的自由软件，不但在类 UNIX 系统中得到了广泛的使用，在苹果电脑的 Mac OS 系统中也是标准编译器。GCC（尤其是其中的 C 语言编译器）已经成为跨平台编译器的事实上的标准。其不只是具有跨平台特性，而且编译程序的效率与普通的编译器相比平均性能要高出 20%～30%。

GCC 可以将 C/C++语言代码及汇编语言程序编译为可执行程序。其实 GCC 是最适合初学者使用的 C/C++编译器。因为通过直接使用 GCC 来编译代码，可以清楚地知道将源代码编译为执行程序的全过程，如连接、编译等。这些过程如果使用成熟的 IDE 环境，是无法体会得到的。虽然直接在命令行使用 GCC 可能无法完成复杂的大型项目（代码量超过 10 万行），但对于初学者来说，是不可能从大型项目起步学习编程的。要写一个简单的二分法检索程序或简单的排序程序使用 GCC 是简洁而又高效的。

2．GCC 编译过程

GCC 由 C 语言源代码文件生成可执行文件的过程，一般来说分为四个阶段：预处理（也称预编译，Preprocessing）、编译（Compilation）、汇编（Assembly）和连接（Linking）。这四个阶段分别由 GCC 调用不同的程序来实现。

- 预处理是调用 cpp 程序来实现的，预处理主要是对包含语句、宏定义等进行处理。
- 编译是调用 cc 来实现的，它是源文件变为后缀为.o 的目标文件的过程。
- 汇编过程是对汇编语言的操作，通过调用 as 来实现，汇编同样也生成目标文件。
- 连接是通过调用 ld 来实现的。这是最为关键的步骤，在该阶段，ld 会将各程序的执行代码放置在程序的适合位置，同时，程序调用的库函数也会以适当的方法设置调用接口。关于连接程序的细节，可查阅相关的 elf 文件格式资料来理解。

GCC 编译时，如果在命令行没有指定其他参数，则 GCC 会完成编译、连接的过程，然后在当前目录中生成一个名为 a.out 的可执行文件。一般来说，Linux 并不依靠文件的后缀来识别文件是否为可执行程序，其判断文件是否可执行依据的是文件的属性。如果对于当前用户而言，文件有运行权限，则该文件为一个可执行程序。

对于 GCC 而言，判断文件的类型主要是依据文件的后缀名。所以，要使用 GCC 编译代码，正确使用文件后缀是保证编译正确的前提。如果写了一个 C++代码，却使用了.c 为后缀，GCC 在编译该文件时会将该文件按照 C 语言代码来处理，一般情况下，编译时会出现错误提示。下面列举出一些 GCC 常用的文件的后缀名，以便在使用时参考。

- c：C 语言源代码文件。
- a：由目标文件构成的档案库文件。
- .C、.cc 或.cxx：C++源代码文件。
- h：头文件。
- i：经过预处理的 C 源代码文件。
- ii：经过预处理的 C++源代码文件。
- m：Objective-C 源代码文件。
- o：编译后的目标文件。
- s：汇编语言源代码文件。
- S：经过预编译的汇编语言源代码文件。

17.1.2　第一个 C 程序

虽然 GCC 是 GNU 的基础，但是，由于现在 Linux 系统正在努力向普通用户的桌面扩展其市场份额，所以 Ubuntu 系统虽然自带了 GCC，但由于并未安装编译所需要的头文件。所以一个新安装的 Ubuntu 系统并不能马上就进行 C 语言编程的工作，还需要的准备工作，就是安装一个名为 build-essential 的软件包。简单地在终端运行如下命令即可完成安装：

```
sudo apt-get install build- essential
```

该软件包依赖于 libc6-dev、libc-dev、g++、make、dpkg-dev 这些编程必须的软件包。但 apt 在安装时会自动处理依赖关系，所以会自动安装这些软件包。

安装这些包后，就可开始进行第一个简单的 C 程序的编制工作了。由于 GCC 需要在命令

行使用，所以我们以在终端操作为例进行介绍。同时，鉴于 vi 的使用应该是 Linux 中开发人员的必用技能，所以本次的编辑器选择了 vi。下面是一个实际使用 GCC 进行编译的全过程。

（1）打开终端，建立一个 C 语言编程的专用目录。

```
mkdir c_code                          #建立名为 c_code 的目录
cd c_code                             #进入该目录
```

（2）使用 vi 新建一个后缀为.c 的 C 语言源程序。

```
vi myfirst.c
```

（3）进入 vi 后，按 i 键，进入插入模式，然后输入以下代码：

```
#include <stdio.h>
int main()
{
  printf("hello gcc\n");
  return 0;
}
```

这是一个极其简单的 C 语言代码。其功能是在屏幕上输出一行内容为 hello gcc 的字符。

（4）这是第一个 C 语言代码的编译工作，所以使用最简单的编译方法。命令如下：

```
gcc myfirst.c
```

使用不带参数的 GCC，会自动完成编译、连接的全过程，同时在当前目录生成一个名为 a.out 的可执行程序文件。输入该指令后，如果屏幕上没有任何提示，然后又回到了命令提示符，则说明程序代码符合语法，编译成功。如果出现错误提示，一般会给出第几行有问题，就去检查代码的相应行，修正错误，然后再运行 GCC 编译。

（5）编译通过后，就可以简单地使用 a.out 调用该程序。使用命令如下：

```
./a.out
```



这里不能直接使用 a.out 来调用该程序，因为 shell 在寻找可执行程序时，并不在当前目录中寻找，所以必须用“./”来显式指定该文件的路径是当前目录。

（6）程序的输出结果如下：

```
hello gcc
```

17.1.3 GCC 的具体编译过程

上例是调用 GCC 自动实现了预处理、编译、连接这几个步骤。为了更清楚地了解 GCC 编译的实际过程，下面将这几个过程分解开来进行介绍。

1. 预处理

预处理的工作主要是处理#include 语句等，然后生成 myfirst.i 文件。如果不加参数“-o myfirst.i”，则会将预处理的结果输出到标准输出。下列的命令的功能是预处理程序文件 myfirst.c。

```
gcc -E myfirst.c -o myfirst.i         #使用 gcc 的 E 参数进行预处理
cpp myfirst.c -o myfirst.i            #使用 cpp 命令进行预处理
```

这两条命令处理的结果是完全一样的，因为 GCC 使用-E 参数实际也是调用了 cpp 指令。预处理生成的 myfirst.i 文件的部分内容如下：

```
# 1 "myfirst.c"
# 1 "<built-in>"
# 1 "<命令行>"
# 1 "myfirst.c"
# 1 "/usr/include/stdio.h" 1 3 4
# 28 "/usr/include/stdio.h" 3 4
# 1 "/usr/include/features.h" 1 3 4
# 330 "/usr/include/features.h" 3 4
…
```

这些代码是给下一步编译连接做准备的，普通用户可以不用理解其意义。

2．生成汇编代码

有时为了某些需要，可以将 C 代码或预处理后的 i 文件汇编为汇编语言代码。分别使用以下命令即可。

```
gcc -S myfirst.c -o myfirst.s       #将 C 代码汇编为汇编语言代码
gcc -S myfirst.i -o myfirst.s       #将 i 文件汇编为汇编语言代码
```

这两条命令生成的汇编代码内容是一样的，生成的汇编代码如下：

```
    .file   "myfirst.c"
    .section    .rodata
.LC0:
    .string "hello gcc"
    .text
    .globl main
    .type   main, @function
main:
    leal    4(%esp), %ecx
    andl    $-16, %esp
    pushl   -4(%ecx)
    pushl   %ebp
    movl    %esp, %ebp
    pushl   %ecx
    subl    $4, %esp
    movl    $.LC0, (%esp)
    call    puts
    movl    $0, %eax
    addl    $4, %esp
    popl    %ecx
    popl    %ebp
    leal    -4(%ecx), %esp
    ret
    .size   main, .-main
    .ident  "gcc: (Ubuntu 4.3.2-1ubuntu11) 4.3.2"
```

```
        .section    .note.GNU-stack,"",@progbits
```

学过汇编的读者可能会发现，这与普通的汇编语言有点不同。其实，这是 AT&T 格式的汇编语言，也被用于很广泛的用途。

3．编译

由汇编代码或 C 源代码生成目标文件，可以用如下任意一种方法。

```
gcc -c myfirst.c -o gccfile.o
gcc -c myfirst.s -o gccsfile.o
as myfirst.s -o asfile.o
cc -c myfirst.c -o ccfile.o
```

这些代码的功能是一样的，均是生成名为后缀为.o 的目标文件。示例为了区分编译时使用的不同方法，将其输出为不同名称的.o 文件，其实这些.o 文件内容是完全一样的。

4．连接

生成后缀为.o 的目标文件后，最后一道工序是将其连接为一个可执行程序。执行如下命令可将编译的.o 文件生成名为 myfirst.out 的可执行程序：

```
gcc gccfile.o -o myfirst.out
```

这样，就可以以 gccfile.o 为基础生成 myfirst.out 可执行程序文件。要运行这个程序，可以使用如下命令：

```
./myfirst.out
```

程序的输出结果与前面直接编译的结果是相同的。

使用 file 检查输出的可执行文件 myfirst.out 类型，得到的结果应该如下：

```
    myfirst.out: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for
GNU/Linux 2.6.8, dynamically linked (uses shared libs), not stripped
```

说明这是一个 ELF 格式、32 位的可执行程序。可运行在 Intel 80386 系列架构的微机，运行的操作系统是 GNU/Linux，使用了动态链接技术。

17.1.4　GCC 高级技巧

GCC 的使用可以是极其复杂的，其自身附带的参数有上百个。支持的系统架构也有几十种，下面是一些稍微复杂的用法。

1．指定头文件位置

在实际的编程工作中，会有头文件与源文件存放在不同的目录的情况。这时，在程序中使用#include "defnumber.h"可能就因为无法找到头文件而宣告编译失败。解决问题的方法有两个：一是在源代码中指定头文件的绝对路径，如“#include "/home/myname/code/header/number.h”。这种解决方法简单直观，但是缺乏灵活性。如果更改了编译环境，或者源代码数量很大，要逐一修改包含头文件，是一件很繁杂的工作；二是在编译时使用“-I”参数来指定头文件的路径。实际工作中，一般选择第二种方法。

使用第二种方法将 resolv.c 编译为 resolv.out 的命令如下：

```
gcc resolv.c -I /home/myname/code/header -o resolv.out
```

这样编译时，编译器就会在-I 指定的目录中寻找头文件。如果找不到，就会依照代码指定的方式，在代码所在目录或系统目录中寻找头文件。如果头文件分散在不同的位置，还可以使用多个-I 参数来指定多个目录。像这样：

```
gcc resolv.c -I /home/myname/code/header -I /home/myname/zd1211rw -o resolv.out
```

该命令指定编译器在/home/myname/code/header 目录与/home/myname/zd1211rw 目录中寻找头文件，并将 resolv.c 编译为 resolv.out。

2．定义符号常量

有时在代码中，会使用一些符号常量来控制程序的行为，如：

```
#ifdef  WIRELESS_ZD1211
     do some thing
#else     /* -----  not WIRELESS_ZD1211  ----- */
     do some thing
#endif    /* -----  not WIRELESS_ZD1211  ----- */
```

这样的话，就有一个问题，怎么来检验定义了符号常量与未定义符号常量的区别。这个问题可以在代码中定义该常量来强制定义该符号常量，但修改代码有与前面所述头文件有着同样的困境。所以，一般应该在编译时使用-D 参数来指定符号常量，命令如下：

```
gcc -D  WIRELESS_ZD121 -I /home/myname/zd1211rw resolv.c -o resolv.out
```

该命令强制定义了符号常量 WIRELESS_ZD121，其余参数与前所述相同，就不再介绍了。

3．定义警告级别

程序员是一群理想主义者，他们均希望编写的代码能按照理想情况运行。但实际情况是代码一般均会出现各种错误或不按理想运行的状态。计算机会检测到所有的错误与可能产生错误的地方。检测到比较大的错误，编译器会停止编译，然后报告出错信息。但还有一些错误信息，其错误程度较低，编译器并不停止编译，只是简单地输出一个警告信息，提示该错误可能会导致的风险。

GCC 可以检测到数量丰富的诊断信息。但默认情况下，许多出错信息是被屏蔽了的，并不显示出来。不过，可以在编译时指定警告信息显示的层次，以显示自己需要的信息。最常用的警告选项是-Wall 选项。其意思是 warning all。使用如下命令显示特定警告信息来编译 test.c，代码如下：

```
gcc -Wall test.c -o test
```


Wall 选项相当于同时使用了这些选项：unused-function、unused-label、unused-parameter 等。使用 Wall 开启了这些警告信息，当然也可以使用-Wno-unused-function 来关闭 unused-function 等特定选项。但是，-Wall 并不像其名字暗示的那样显示所有警告信息，下面是-Wall 选项没有开启的警告项。

- sign-compare：将有符号类型变量和无符号类型变量进行比较时发出警告。
- missing-prototypes：如果没有预先声明函数原形就定义了全局函数，编译器就发出警告。即使函数定义自身提供了函数原形也会产生这个警告。这样做的目的是检查没有在头文件中声明的全局函数。
- unreachable-code：如果发现从未执行的代码时给出警告。

- Inline：如果某函数不能内嵌（inline），则即使是声明了 inline 项，编译器也将发出警告。
- disabled-optimization：当某项优化选项由于其需要过长时间或过多资源而导致不能完成时，给出警告。
- Werror：有任何的警告信息则直接停止编译过程。这个选项在编制对于质量要求高的场合很有用。

4．其他 GCC 的选项

前面介绍了 GCC 的一些选项，但是 GCC 的参数是极其复杂的。本小节再来介绍一些 GCC 的常用参数。一般来说，GCC 命令的完整用法如下所示：

```
gcc [option] [filename]
```

其中 option 是定义编译器行为的参数，filename 是指定给编译器操作的文件名。option 可以取如下参数。

- -c：只将源代码编译为以.o 为后缀的目标文件，而不是直接将这些文件连接成可执行文件。通常用于编译不包含主程序（main 函数）的子程序代码。
- -o filename：指定输出文件名。如果不使用这个选项，GCC 会将文件输出为 a.out，而不管输出文件的类型。
- -g：产生调试所需的符号信息。如果要使用 GDB 进行调试，则必须使用该选项。但是该选项所产生的调试符号，并不能被其他调试器识别。
- -O（大写字母 o）：在程序的编译、连接过程中，对代码进行优化。相对来说，编译时使用该选项，会根据编译机器架构与操作系统进行相应的优化。所以，在开源世界中，下载源代码并自己进行编译会得到比直接使用在其他机器上编译的程序更好的性能。不过，编译时使用该选项，会降低编译的速度。
- -O2：比-O 更好的优化编译、连接。现代算法已经进化到很智能的程序，使用编译器的优化参数可以获得的性能提升很多时候比人工优化性能更好。
- -L dir：在程序连接时，会首先在 dir 目录寻找类库文件，然后在系统预设路径中寻找类库。也可以使用多个-L 参数指定多个类库存放的目录。
- -lname：在程序连接时，加载名为 libname.a 的函数库。如-lmath 表示连接名为 libmath.a 的数学函数库。
- --sysroot=dir：让目录 dir 作为头文件和库文件的根目录。
- -B dir：将目录 dir 添加到编译器的搜索路径中。
- -b：指定 gcc 编译的目标机器架构，如 ARM、DEC，GNU/Linux（-muclibc）、IA64、MIPS 等，这也可以看出 gcc 的超级跨平台性。
- -V n1.n2：使用 n1.n2 指定版本的 gcc 进行编译。
- -E：仅作预处理，不进行编译等工作，生成 i 文件。
- -S：将源代码编译为汇编语言代码。
- -x lang：将代码依照特定语言进行编译，可以取的值包括但不限于 C、C++、Objective-c、Objective-c++、Assembler、Ada、Java 等。在一系列的编译中，该选项会一直保持，直到另一个-x lang 出现，指定另一种编译语言。如果取值为 none，意味着让 GCC 依据文件的后缀来自动判断语言种类。

17.2　makefile 简介

makefile 是一种自动处理编译中代码与产生程序关系的文件，其可以自动判断代码文件与生成程序的新旧程序，可以执行许多复杂的条件判断生成相应不同的程序或完成不同的工作。可以说，编写高效的 makefile 文件是 Linux 环境中 C/C++程序编写的基础技能。本节简单介绍了 makefile 文件编写方法。

17.2.1　第一个 makefile 文件

由 17.1 节的介绍可以看到，GCC 有数量极其庞大的选项，而且对于源代码来说，编译又可以分为许多步骤。对于小型项目，每一个文件进行一种编译步骤还是可以忍受的。但对于稍微复杂些的项目，要对成百上千的文件进行各种复杂的编译操作，需要的时间暂且不论，各种复杂关系的处理就是一项难题，稍微有些疏忽就会出现不可预知的错误。

1．makefile 简介

这时，就需要一种可以进行自动编译操作的工具来实现这些目标了。在 Linux 甚至在 Windows 环境中，完成复杂的编译工作的文件都被称为 makefile。

makefile 对于许多 Windows 程序员来说，显得有些陌生。因为，Windows 的编程 IDE 与其系统一样，隐藏了许多实现的细节。也许有许多人使用过 Windows 中一款著名的 C 语言编程 IDE Visual C++，但并没有接触到任何关于 makefile 的概念，也许大多数使用者甚至都不知道 VC++使用的是哪一种 C 编译器。而实际上，在 VC++中，处理代码的编译同样也使用了 makefile 文件。

Linux 的风格与 Windows 的截然不同，其暴露了许多系统级的操作细节，对编程也不例外。前面介绍的 GCC 可以让使用者体会到将代码变为可执行程序的全过程。而本节介绍的 makefile 会让读者知道那些 IDE 是如何处理复杂的代码编译关系的。

其实 makefile 的原理很简单，就是一个名为 makefile 或 Makefile 的文本文件，指定了一些自动执行的命令和编译程序的规则。这些命令与规则由一个名为 make 的程序处理。make 会根据 makefile 的规则生成相应的文件。

2．编写第一个 makefile

makefile 的文件可以极其复杂，不过，本着由易及难的理念，还是以一个简单的 makefile 来进行介绍。假设在～/c_code 目录中有一个名为 myfirst.c 的 C 源代码文件，下面是给该文件编写 makefile 的步骤。

（1）改变当前目录到 C 源代码所在的目录，使用如下命令。

```
cd ～/c_code
```

（2）新建一个名为 makefile 的文件。

```
vi makefile
```

（3）在其中加入以下内容，这就是第一个简单的 makefile，但是毫无疑问，它是可以正常工作的。

```
#第一个 makefile 的内容
myfirst : myfirst.o
```

```
	gcc myfirst.o -o myfirst
myfirst.o : myfirst.c
	gcc -c myfirst.c -o myfirst.o
```

（4）保存该文件退出后，运行 make 命令完成源代码的编译工作。输出内容应该类似这样：

```
make
Compilation started at Sat Dec  6 22:07:19
gcc -c myfirst.c -o myfirst.o
gcc myfirst.o -o myfirst
Compilation finished at Sat Dec  6 22:07:20
#这就说明所有的规则已经得到处理，然后依照规则生成了名为 myfirst 的可执行文件
```

（5）执行编译生成的程序，使用如下命令。

```
./myfirst
#执行编译生成的程序，会得到一行内容是“hello gcc”的输出
```

在这个过程中，并未调用 gcc 或 cc 之类的编译器，但在当前目录中却生成了相应的目标文件与可执行文件。这就是 makefile 的功劳。

3．makefile 格式分析

下面我们来详细解析一下刚才编写的简单的 makefile 文件。

第一行：“myfirst : myfirst.o”。

makefile 文件中出现的第一个目标（除了变量声明与注释之类的内容）是 makefile 的最终目标，指明了 makefile 最后应该生成什么文件。本例的最终目标是生成 myfirst 文件，同时这一行也被称为是目标行，因为它指定了生成的目标。myfirst 后面有一个冒号，冒号后面的内容是生成 myfirst 的前提。也就是说要生成 myfirst，必须要有一个名为 myfirst.o 的文件存在。

第二行：gcc myfirst.o -o myfirst。

第二行其实是第一行内容块的一部分，说明了如何由 myfirst.o 生成 myfirst。这一行是由 Tab 键产生的制表符开头的。以 Tab 开头的语句被称为命令行。命令行的内容是可以在 Shell 中实际执行的命令。而且命令行必须以 Tab 开头。

4．makefile 中的模块

示例 makefile 文件第一行与第二行称为 makefile 的一个模块。由第一行指定目标，然后由命令行生成目标。注意，命令行必须以 Tab 键开头，以告诉 make 程序：“这是一个命令行，请执行该行的代码”。可以同时指定多个目标与依赖条件，也可以在一个模块内执行多个命令。如：

```
target1 target2 target3 : prereq1 prereq2
	comand1
	comand2
	comand3
```

理解了第一行一与第二行的内容，则其余行的内容也就很容易理解了。第三行指定了目标是 myfirst.o，依赖于 myfirst.c，第四行指定了由 myfirst.c 生成 myfirst.o 的方法。

5．makfile 的处理规则

make 命令在处理 makefile 时是递归处理的。本例中，先遇到的目标是 myfirst，其依赖是 myfirst.o，但此时目录中并无 myfirst.o 文件，由于并未处理完整个 makefile 文件，所以 make 并不报错，只是继续向下运行，就发现了目标 myfirt.o，其依赖于 myfirst.c。此时，make 会去检查目录，看是否有 myfirst.c 文件。如果有该文件，就使用相应的命令生成 myfirst.o，然后再用 myfirt.o 生成 myfirst 文件。这个过程，通过查看 make 命令的输出也可以看到。

同时，make 在处理 makefile 时会检测目标文件与依赖文件的时间戳，如第一行 myfirst 与 myfirst.o。如果 myfirst.o 比 myfirst 文件要新的话，就会重新生成 myfirst 文件，否则，make 会忽略掉这个模块。这个特性降低了编译文件时的时间开销，因为其只增量编译更新过的文件。还有一点要注意的是，makefile 文件必须以 makefile 或 Makefile 为名。

17.2.2　对简单 makefile 文件的扩充

makefile 文件就像是一种小型的脚本语言，所以其也支持变量的定义，而灵活使用变量，可以增强 makefile 的适应性与灵活性。下面是一个使用变量的 makefile。代码中，有详细的注释。

```
#这个脚本会使用两种方法进行编译、连接，一种是将代码通过编译为汇编代码来实现，另一种是直接编译
##############################
NAME = myfirst
cc = gcc
ac = as
CFLAG = -Wall -O1 -g
#这是编译源程序的编译选项，具体含义可参见前面 gcc 参数的介绍
${NAME} asfile : ${NAME}.o asfile.o
#使用变量时，应该使用$提取符，然后用大括号将变量名括起来
    ${cc} ${CFLAG} ${NAME}.o -o ${NAME}
    ${cc} ${CFLAG} asfile.o -o asfile
 ${NAME}.o : ${NAME}.c
    ${cc} -c ${NAME}.c -o ${NAME}.o
 asfile.o :  ${NAME}.s
    ${ac}  ${NAME}.s -o asfile.o
#由汇编代码生成目标文件
 ${NAME}.s :  ${NAME}.c
    ${cc} -S  ${NAME}.c -o  ${NAME}.s
#生成汇编代码的方法
other : ${NAME}.o
#other 选项并未出现在最终目标中，所以直接使用 make 命令不会执行这一行。要执行这一行，必须使用 make other 来执行
    ${cc} ${CFLAG} ${NAME}.o -o other
#这里并未使用显式规则来指定${NAME}.o 的生成方式，因为对于 make 命令而言，如果在规则中发现 name.o 文件，其会自动寻找同名的 c 代码(name.c)，然后自动根据找到的代码调用相应的编译器来编译生成 name.o 文件
```

17.2.3 在 makefile 中使用函数

makefile 中，可以使用许多函数来简化操作。这些函数包括字符处理函数、目录、文件名函数等。makefile 调用函数的方式是$（函数名 参数 1，参数 2）或${函数名 参数 1，参数 2}。

1．字符串替换函数 subst

$(subst source,target,orderstring)其作用是将 orderstring 中的 source 字符串替换为 target 字符串。

如：${subst my,me,ThisIsMyWorld}，将 ThisIsMyWorld 中的 my 替换为 me。替换后字符串的内容是 ThisIsmeWorld。

2．模式替换函数 patsubst

$(patsubst source,target,orderstring)，看起来和 subst 很像。不过其支持简单的正则表达式，可以使用%来替代字符串。

如：$(patsubst %.o,%.c, main.c class1,c class2.c)，会将“main.c class1,c class2.c”中所有结尾的.c 替换为.o，替换后的结果是“main.o class1,o class2.o”。

3．过滤函数 filter

$(filter PATTERN...,TEXT)，匹配 TEXT 中所有符合 PATTERN 的字符串。

如：$(filter %.c %.s,main.c assem1.s makefile readme.txt)，会将“main.c assem1.s makefile readme.txt”中以.c 与.s 结尾的字符匹配出来。匹配的结果是“main.c assem1.s ”。

4．取文件名函数 notdir

$(notdir NAMES...)会取出 NAMES 中不是以“/”结尾的字符串。

如：$(notdir myfirst.c myfirst.o debug/ release/)，会取出"myfirst.c myfirst.o"。

5．取后缀函数 suffix

$(suffix NAMES...)会取出 NAMES 中的所有后缀。

6．取文件名函数 basename

$(basename NAMES...)会取出 NAMES 中的除去后缀的内容。

7．添加后缀函数 addsuffix

$(addsuffix SUFFIX,NAMES...)，会在 NAMES 中字符串后添加 SUFFIX 后缀。

8．匹配特定字符

$（wildcard PATTERN）列出当前目录中所有符合 PATTERN 模式的字符串。

如$（wildcard *.c），会列出当前目录中所有以.c 为后缀的字符串。

初步了解了这些函数后，就可以在 makefile 中使用这些函数了。通过一个例子，就可以看到，灵活使用会简化 makefile 的编制。

```
#一个简单的使用函数的 makefile 例子
NAME = myfirst
ccc = gcc
CFLAG = -Wall -O1 -g
#这是编译源程序的编译选项，具体含义可参见前面 gcc 参数的介绍
objects := $(patsubst %.c,%.o,$(wildcard *.c))
func : $(objects)
      ${ccc} ${CFLAG} -o ${NAME} $(objects)
```

$(patsubst %.c,%.o,$(wildcard *.c))看起来有点复杂。不过，参照上面的函数定义，可以很容易理解。$(wildcard *.c)是得到当前目录中所有以.c 为后缀的文件列表。patsubst 将 wildcard 取出的 C 代码文件名中的.c 替换为.o。本例中会将 myfirst.c 替换为 myfirst.o。在本例中看起来似乎不够强大，但如果目录中有上百个 c 代码文件，这个 makefile 仍然可以正常工作，那就是 makefile 的强大之处。

17.2.4　清理 make 过程中的中间文件

编译完成后，清理掉那些没有用的中间文件是个良好的编程习惯。一般情况下，清理中间文件可以这样进行：

```
.PHONY: clean
clean:
    - rm *.o
```

可能.PHONY: clean 看起来有些奇怪，其实这是个很有用的 makefile 技巧。因为如果不使用该行，则可能在使用“make clean”会出现某些问题。如果当前目录中没有名为 clean 的可执行程序，则“make clean”会正常执行，但是，如果当前目录中一个可执行程序名为 clean，运行 make clean 会运行名为 clean 的可执行程序，而不是运行 makefile 中的清理命令。

使用.PHONY: clean 将 clean 定义为一个伪目录.PHONY 的依赖，这样使用 make 运行时，就不再试图在当前目录中查找名为 clean 的程序，不管当前目录中有无 clean 程序，均会执行相应的清理命令。同时，定义了伪目标也会在一定程度上提高 makefile 的性能。

使用前面有短线的 rm 指令，则是忽略该操作的错误与警告信息。有时，这些中间文件可能被另一个程序调用，无法删除，rm 命令会返回一个错误信息。使用- rm 会忽略这些警告信息，继续执行其他代码。一个复合了本节所介绍内容的 makefile 应该类似这样：

```
#这个脚本会使用三种方法进行编译、连接，一种是将代码通过编译为汇编代码来实现，另一种是直接编译
#第三种方法是使用函数来获得当前所有 c 语言文件，然后自动根据隐含依赖关系生成目标文件
#本例还定义了 clean 指令，用来清理在编译过程中产生的目标文件(.o 文件)
NAME = myfirst
ccc = gcc
ac = as
CFLAG = -Wall -O1 -g
#这是编译源程序的编译选项，具体含义可参见前面 gcc 参数的介绍
objects := $(patsubst %.c,%.o,$(wildcard *.c))
${NAME} asfile : ${NAME}.o asfile.o
    ${ccc} ${CFLAG} ${NAME}.o -o ${NAME}
    ${ccc} ${CFLAG} asfile.o -o asfile
  ${NAME}.o : ${NAME}.c
    ${ccc} -c ${NAME}.c -o ${NAME}.o
  asfile.o :  ${NAME}.s
    ${ac}  ${NAME}.s -o asfile.o
  ${NAME}.s :  ${NAME}.c
    ${ccc} -S  ${NAME}.c -o  ${NAME}.s
```

```
func : $(objects)
    ${ccc} ${CFLAG} -o ${NAME} $(objects)
.PHONY: clean
clean:
- rm *.o
```

该 makefile 有如下的使用方法。

- make：处理默认的最终目标，会生成 myfirst、asfile 可执行程序文件。
- make func：处理 func 定义的模块。只生成 myfirst 可执行文件。
- make clean：清理编译过程上的.o 文件。

17.3 程序调试

程序在编写完成后，由于程序员的经验不足、或是由于输入错误，程序往往并不会完全依照程序员的理想来运行，很可能会出现各种各样的错误。这时，就需要检查源代码或者对程序进行调试，查看其每一步的输出，以弄清问题的根源。并修改相应的源代码来解决问题。本节将简单介绍在编程中经常出现的一些错误提示及解决方法。并且还会简单介绍一款强大的调试器 GDB 的使用方法。

17.3.1 GCC 编译的几种错误信息及其解决方法

GCC 编译中可能会得到一些错误，这些错误会导致编译过程的中止。不过，好在 GCC 同时也给出了详细的出错信息以供分析查看，这样在出错时，就不会手足无措了。一般情况下，GCC 的编译出错可分为四个类型，下面分别进行介绍。

1．语法错误

这是初学者最容易犯的错误，一般是在输入代码时括号不匹配或者是用错了关键字。如使用下面的代码：

```
#include <stdio.h>
int main()
{
     for ( int i=1;i<100 ;i++ ) {
     printf("hello gcc\n");
     return 0;
}
```

然后使用上节介绍的 make 文件进行编译，这时编译器会给出如下提示：

```
gcc -c myfirst.c -o myfirst.o
myfirst.c: 在函数‘main’中:
myfirst.c:4: 错误： 在 C99 模式之外使用‘for’循环初始化声明
myfirst.c:7: 错误： expected declaration or statement at end of input
make: *** [myfirst.o] 错误 1
gcc -S  myfirst.c -o  myfirst.s
myfirst.c: 在函数‘main’中:
```

```
myfirst.c:4: 错误： 在 C99 模式之外使用‘for’循环初始化声明
myfirst.c:7: 错误： expected declaration or statement at end of input
make: *** [myfirst.s] 错误 1
make: 由于错误目标“myfirst”并未重新创建。
Compilation exited abnormally with code 2 at Sun Dec  7 14:03:48
```

这些错误信息的意思是语法错误可能从第 4 行就开始了。读者可以自行判断一下这句代码有无问题，其实单就这一行而言，语法是没有问题的。但是再向后看到第 7 行，是一个右大括号，对于编译器来说，问题就出来了，这个大括号是属于 for 循环，还是属于主函数呢？由于语法不完整，编译无法继续进行，只能给出错误信息，然后就退出了。

知道问题的根源后，简单地在 printf 语句后加上一个右大括号使其与 for 循环的大括号匹配，语法就没问题了。编译就会得到想要的结果。遇到语法错误，就可以先检查错误提示中出现的第一个行号，如果该行没有问题，就检查该行所开始的语法模块结构是否完整，然后修正该结构，一般会解决语法问题。

2．头文件错误

如果编译给出的错误提示说 Can not find include file xxxx.h，就说明是指定的包含文件有问题，系统在编译过程中找不到指定的头文件。如果该头文件不存在，就应该创建该头文件。但是如果是头文件存在，而只是编译器找不到的话，就应该使用-I 参数来指定该头文件所在的目录。

3．类库错误

如果出现类似“ld: -lm: No such file or directory”的错误，可能是在默认的目录内找不到相应的类库。这种问题的解决方法是在编译时使用-L 参数指定要使用的类库所在的目录。

4．未定义符号

出现类似 Undefined symbol 的提示，说明在编译过程中发现了没有被定义的符号变量。这个错误产生的原因可能是定义的该符号的代码没有被编译，或者是包含该符号的类库没有被加载，可根据不同的情况采取不同的处理方法。

上述的几个错误，从某种意义上来说是静态的错误，也是比较外在的错误。一般来说，可以很容易地在编译时解决这些问题。而真正的程序调试，一般是指编译过程并未出现错误，但是实际输出结果却与程序编制的初衷不相符。这种问题是由于程序算法设计上的问题，需要使用动态的调试技术来发现错误。动态调试程序，对于许多用户来说，首选 GDB。本节后面的内容，就将介绍 GDB 调试程序的一般过程。

17.3.2　GDB 简介

GDB 是 Linux 平台一款功能极其强大的调试器，全称是 The GNU debug，是 GNU 软件库的一种。GDB 是一款命令行的调试器，支持 C/S 结构，可用于远程调试及调试嵌入式系统板。由于其功能比较复杂，这里只介绍对于初学者来说比较实用的几条命令，有兴趣深入探究 GDB 的读者可以阅读 Richard M. Stallman 与 Roland H. Pesch 所著的《Debugging With Gdb: The Gnu Source-Level Debugger Fifth Edition，for Gdb Version》一书。

GDB 可以调试的前提是，被调试程序在编译时必须加上 g 参数。否则在调试时由于无法添加断点，所以无法让程序在指定的位置中断，会导致无法进行调试。GDB 的使用方法很简单，在终端中输入如下所示的任意一条命令即可。

```
gdb 程序名
gdb
```

这两种方式均可进入 GDB 的交互式调试界面。如果在命令行指定了程序名，则 GDB 会自动加载该程序。如果命令行不指定程序名，则需要在进入 GDB 交互式界面后自行加载程序进行调试。GDB 的交互模式是 GDB 的主要功能区，是用来完成各种调试功能的界面。在交互模式中可以使用许多命令，现将常用的命令列于下。

- file：加载要调试的程序。
- kill：终止正在调试的程序。
- list：列出 10 行程序的源代码。
- next：单步执行程序。
- step：单步执行程序。与 next 不同的是，其会进入调用的函数内部。而 next 只需要调用函数的结果。
- run：运行加载的程序。
- quit：退出 GDB。
- watch：监视一个变量的值。
- break：在代码里设置断点。程序运行到断点处时会停下来，然后用户可用 next 或 step 单步运行程序。但使用 break 的前提是程序在编译时使用了 g 参数。
- make：不用退出 GDB，重新编译代码，然后在 GDB 中运行。
- shell：可调用 Shell 命令。
- bt：查看函数堆栈。
- c：继续运行。
- finish：退出函数。
- info：查看相关信息，如 info break。

17.3.3 GDB 例程

为了说明 GDB 的功能，下面以一个简单的代码为例来说明调试的一些基本操作。比如，为了实现输出一个由“*”组成的三角形，编写了如下代码：

```
//filename:gdb4.c
#include <stdio.h>
int main()
{
  int i=0,j=0;
  for(;i<10;i++)
    {
    for(;j<i;j++)
      {
      printf("*");
      }
    printf("\n");
    }
  return 0;
 }
```

然后使用“gcc -Wall -O1 -g -o gdb4.out gdb4.c”编译该文件得到名为 gdb4.out 的可执行程序。但实际执行该程序后，输出并不是想象中的一个三角形，而是简单地输出了一列*。所以，使用 GDB 来检查到底发生了什么，以致输出偏离了程序设计的初衷。下面是使用 GDB 调试的步骤。

（1）运行 GDB 程序，使用如下命令。

```
earthfans@mylaptop:～/tmp$ gdb gdb4.out
#gdb 后面跟要调试的 gdb4.out 程序名，开始调试该程序。如果在命令行没有指定调试的目标程序，可以在 gdb 命令行中输入“file gdb4.out”指定调试目标程序为 gdb4.out
#下列信息是欢迎信息
GNU gdb 6.8-debian
Copyright (C) 2008 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i486-linux-gnu"...
#显示 GDB 的状态信息
```

（2）设置程序断点（注意，如下命令均是在 GDB 环境中运行）。

```
(gdb) b main
#设置程序的断点在函数 main 处，完整的命令是 break。程序在运行时会暂时停在设置了断点的行处。如果没有设置断点，程序会自动运行直至结束，这样就无法进行调试了
Breakpoint 1 at 0x80483d7: file gdb4.c, line 3.
#显示状态信息，断点设置在源文件 gdb4.c 中的第 3 行
```

（3）显示源代码，使用命令 l（字母）。

```
(gdb) l
#list 命令，简写为 l。显示源程序文件内容的前 10 行。要使用 list 命令显示源代码，则必须在编译程序时使用-g 选项
1    #include <stdio.h>
2    int main()
3    {
4      int i=0,j=0;
5      for(;i<10;i++)
6        {
7        for(;j<i;j++)
8          {
9          printf("*");
10         }
```

（4）运行程序，使用命令 run 或简单使用 r 均可。

```
(gdb) run
#运行程序，在参数中指定了程序名，只是加载了该程序，但程序其实并未运行，在真正调试前，需要使用 run 或 r 运行已经加载的程序。使用 run 命令后，程序会停留在第一个断点处
Starting program: /home/earthfans/tmp/gdb4.out
```

```
Breakpoint 1, main () at gdb4.c:3
3   {
#程序开始从代码的第 3 行开始运行
```

（5）单步运行程序，使用命令 next 或 n。

```
(gdb) n
#使用 next 命令来执行下一行的代码，还有与之类似的指令是 step(可以简写为 s)。next 指令在执行时如果遇到子函数，只是简单地获得子函数的运行结果。而 step 会进入该子函数，单步运行函数内的代码
11      printf("\n");
#第三行代码执行后，程序流程转移到第 11 行。程序调试的一个很重要的内容是分析程序运行的流程，应该详细分析及了解程序的流程，程序的流程是代码的逻辑结构，也是检查程序逻辑错误的重点
#这里代码转移到第 11 行，是因为 i 与 j 均为 0，所以不执行 j 循环中输出“*”的第 9 行代码
```

（6）继续单步运行程序。

```
(gdb) n
5     for(;i<10;i++)
#这里开始第二个循环
```

（7）显示特定变量的内容，可以使用命令 print 或 p。

```
(gdb) print i
#显示变量 i 的内容
$1 = 0
#现在，i 的值是仍然是 0.
```

（8）继续单步运行程序。

```
(gdb) n
#执行完第 5 行后，i 的值变为 1，所以进入了 j 循环，执行输出“*”的第 9 行
9         printf("*");
(gdb) n
#在第 7 行的循环体中进行判断
7       for(;j<i;j++)
```

（9）继续单步运行程序。

```
(gdb) n
#由于 i 也增加了 1，变为 1，不满足 j<i 的判断，所以跳出 j 循环，来到 11 行
11      printf("\n");
```

（10）显示程序运行堆栈，使用命令 bt。

```
(gdb) bt
#显示程序执行堆栈，学习了数据结构的相关内容后，应该很容易理解堆栈的意义
#0  main () at gdb4.c:11
```

（11）继续单步运行代码。

```
(gdb) n
5     for(;i<10;i++)
(gdb) n
```

```
9          printf("*");
#现在是 j 循环
(gdb) n
7       for(;j<i;j++)
#再判断是否 j<i，如果 j<i，则继续输出“*”
#现在的 i 值为 2
(gdb) n
11      printf("\n");
#这儿似乎就有点问题了，i 是 2，应该输出两个“*”，但是 i=2 的 j 循环只运行了一次。那么问题就应该出在这儿了。为什么 i=2 时 j 的循环只运行一次，再仔细观察代码，发现 j 的值只初始化了一次。在每个循环中，j 均增加 1，其值与 i 值始终是同时增加的。所以，每次 j 循环只执行一次
```

（12）找到问题后，修改代码也就简单了。只需在每次 j 循环开始时，赋予 j 一个 0 值即可。具体方法是将代码的第 7 行更改为“for （j=0;j<i;j++)”即可。

（13）重新编译，再次运行，结果就比较理想了。输出结果应该类似这样：

```
*
**
***
****
*****
******
*******
********
*********
```

通过这个例子的学习，读者应该具备了初步的调试代码的能力。



GDB 虽然功能强大，但其并没有良好的图形界面供初学者使用，这也是阻碍初学者的一个主要因素。不过，如果使用 Emacs 进行开发，则可使用相关的插件在使用 GDB 时打开多个窗口，同时显示系统寄存器、堆栈、变量等信息，可以方便调试。

17.4　C/C++开发环境

使用最基本的文本编辑器来代码，虽然可行，但很多时候并不太方便。对于大多数人来说，这样做效率也不太高。所以，一个高效的开发环境对于实际的开发工作是大有裨益的。Linux 的程序员，一般经常使用 vim 或者 Emacs 来搭建开发环境。这两种开发环境的性能各有千秋，可供不同的用户选择。

17.4.1　使用 vim

使用 vim 作为 C/C++的开发环境，前期的配置工作与 vim 的基本用法可以参见第 5 章相关内容。本节介绍的内容主要是如何更高效地书写 C 代码。参照第 5 章完成的配置用来浏览代码是比较高效的，但如果要在 vim 中书写代码，就显得有点不够强大了。在 vim 中书写 c 代码，

最好安装一个叫做 c.vim 的插件。其下载地址是 http://www.vim.org/scripts/download_script.php?src_id=9602。

下载得到一个压缩文件，解压后得到 support、doc、ftplugin、plugin 四个目录与一个说明文件 README.csupport，将这些文件全都复制到～/.vim 目录，然后重新启动 vim 即可完成插件的安装。

如果使用图形界面的 vim，如 gvim。安装了 c.vim 插件后，菜单中就多了名为“C/C++”的一个菜单。先打开该菜单浏览一下菜单的内容，可以看到这个菜单内容很丰富。有给程序添加注释的、有插入一个代码块的，等等。下面是插件菜单项的相关功能。

1.“注释”菜单项

先选中要添加注释的行，然后选择不同的菜单项，则会产生不同的注释。如选择 end-of-line comment，则会产生对齐的注释。效果如下所示：

```
print_double_array ( double array[],            /*  */
                  int    n,                    /*  */
                  int    columns,              /*  */
                  char*  arrayname             /*  */
                  )                           /*  */经
```

使用这项注释的键盘命令是“\cl”。如果要注释代码，也就是说要将代码变为注释，就要选择 code|comment。如果要将代码块前的注释符取消，可以选择 comment|code 命令。date 命令与 date+time 菜单项可以在光标所在位置插入当前的时间与日期。

2.“Statement”菜单项

使用该菜单项中相关内容可以方便地插入 C 语言的循环与判断代码块，如 do while，if，for 等结构。

3.“preprocessor”菜单项

定义了一些预编译的处理命令，如#ifdef #else #endif 等结构。

4.“C++”菜单项

是专门针对 C++的一些操作，如 cin、cout 命令、类的定义等。

5.“Run”菜单项

对于许多用户来说，Run 菜单无疑是最为有用的，里面提供了不用编写 makefile 就可进行编译连接的功能。这样的话，对于初学者来说，减轻了许多麻烦。

6. 插件的快捷键

这些菜单中提供的命令，不只是能在图形界面的菜单中使用。其提供了丰富的组合键操作，也同样可以在文字界面中使用。部分组合键列表如下所示。

- \cl：添加行尾注释。
- \c*：生成 /* */风格注释。
- \c/：生成 //风格的注释。
- \co：取消行的注释。
- \cd：插入日期。
- \ct：插入日期、时间。
- \sd：插入 do { } while 结构。

- \sf：插入 fore 结构。
- \sfo：插入 for { }结构。
- \si：插入 if 结构。
- \sif：插入 if { }结构。
- \sie：插入 if else 结构。
- \sife：插入 if { } else { }结构。
- \sw：插入 while 结构。
- \swh：插入 while { }结构。
- \p<：插入#include <>。
- \p"：插入#include "" 。
- \if：插入函数定义语句。
- \im：插入主函数 main()。
- \+c：插入类的声明。
- \rc：保存编译当前文件。
- \rl：连接当前文件。
- \rr：运行当前文件产生的程序。
- \rm：使用 make 来编译程序。
- \lcs：加载 C/C++支持菜单。
- \ucs：关闭 C/C++支持菜单。

7．使用示例

下面以一个简单的例子来说明其基本操作，以感受该插件带来的编程的快感。具体步骤如下。

（1）在终端中以 gvim test.c 新建一个名为 test.c 的 C 源码。

（2）选择菜单 C/C++|preprocessor|#include std lib|stdlib.h，会看到代码中插入了#include <stdlib.h>。

（3）输入”\im“插入主函数。

（4）输入“\sife”插入 if { } else { }。在其中输入相关的语句。

（5）将光标移动到主函数外，输入\if，插入函数。此时，光标会移到屏幕最下方，需要在该处输入函数名，按下 Enter 键，就生成了一个类型为 void 的函数。现在生成的 C 代码文件如下：

```
#include    <stdlib.h>
    void
byCvim (  )
{
    return ;
}       /* -----  end of function byCvim  ----- */
    int
main ( int argc, char *argv[] )
{
    if ( 3>2 ) {
    printf("3>2");} else {
    printf("ignore");
    }
```

```
    return EXIT_SUCCESS;
}               /* ----------  end of function main  ---------- */
```

（6）输入\rc 编译代码。输入“\rl”链接目标文件。输入“\rr”运行当前代码。应该可以看到在 vim 的屏幕下方出现程序的运行结果。

8．vim 中编程相关的组合键

- [[：跳转到前面第一个“{”。
-]]：跳转到后面第一个“{”。
- []：跳转到前面第一个“}”。
-]]：跳转到后面第一个“}”。
- {：跳转到前面的一个空行。
- }：跳转到下一个空行。
- gd：跳转到当前变量的定义处。
- "：回到搜索开始前的位置。因为搜索改变了位置。
- %：跳转到匹配的括号处。
- Ctrl+p：显示当前输入的自动补全代码列表，可以用 Ctrl+N、Ctrl+P 组合键进行切换。



vim 的 c.vim 插件虽然功能比较实用，但使用方式对于大多数用户来说可能是比较陌生和怪异的。要熟练使用该插件，理解 vim 的几种模式之间的区别是关键所在。理解了 vim 的模式然后多加练习，使用该插件还是有很高效率的。

17.4.2 使用 Eclipse

Eclipse 是 Java 语言编写的可跨平台的一款功能强大的 IDE（开发集成环境）。其开始是作为 Java 的开发平台，但由 IBM 投入 5000 万美元进行技术改造，添加了许多新特性后，已经成为一款支持多语言环境，包括 C/C++以及 php、python 等语言的强大开发环境。虽然其由 Java 语言编写，使得性能有所损失，但由于其丰富的插件支持与强大的功能，还是逐渐获得了市场的认可。本节，我们就来了解一下其强大的功能。

1．安装 Java 支持环境

由于 Eclipse 是基于 Java 技术构建的，所以在运行 Eclipse 之前要确保系统中有一个正常运行的 Java 环境。而 Ubuntu 默认是没有安装 Java 的，所以要先安装 Java。安装的方法很简单，见如下步骤。

（1）在终端中输入如下命令安装最基本的 JRE：

```
sudo apt-get install sun-java6-jre
#安装最基本的 JRE(Java 运行环境)
```

（2）如果还要进行 Java 的开发，可以再安装一个 JDK。命令如下：

```
sudo apt-get install sun-java6-jdk
```

安装中，会在终端中出现是否同意 Java 的使用协议。此时，按 TAB 键切换到 OK，然后回车即同意该协议，然后完成安装。

（3）如果系统中安装有多个 Java 环境的话，可能刚刚安装的 Java 并不会被系统使用，可

以使用如下命令来切换 Java 环境：

```
sudo update-alternatives --config java
```

出现的内容应该类似如下：

```
arthfans@mylaptop:~/tmp$ sudo update-alternatives --config java
[sudo] password for earthfans:
现有 4 个可选项，它们都提供了“Java”<
   选择         可选项
-----------------------------------------------
*         1    /usr/lib/jvm/java-6-sun/jre/bin/java
          2    /usr/bin/gij-4.2
          3    /usr/bin/gij-4.3
 +        4    /usr/lib/jvm/java-gcj/jre/bin/java
要维持默认值[*]，按回车键，或者键入选择的编号：1
使用“/usr/lib/jvm/java-6-sun/jre/bin/java”来提供“Java”
```

在选择时，应该输入包含 "sun" 的行的前面的数字。如上面显示，则应输入 1，然后按回车键确定。

（4）在将当前的 Java 环境切换至新安装的 Java-6-sun 后，还需要对 JAVA 环境变量进行适当的设置，使用下面的命令打开环境变量文件：

```
sudo gvim /etc/environment
```

在其中添加如下两行：

```
CLASSPATH=.:/usr/lib/jvm/java-6-sun/lib
JAVA_HOME=/usr/lib/jvm/java-6-sun
```

2．下载 Eclipse 与 CDT

虽然软件库中有 Eclipse，但其版本过于陈旧，在很多时候已经失去了使用价值。所以推荐在 Eclipse 的主页进行下载，现在的最新版本是 Eclipse3.4.1，其开发代号是 Ganymede。Eclipse 的主页是 http://www.Eclipse.org/。打开网页后，单击页面右上角“Download Eclipse”图标，打开下载页面，如图 17.1 所示。

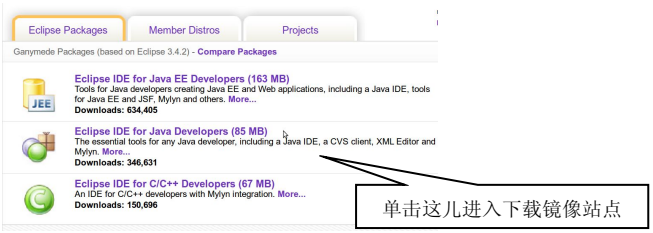


图 17.1　Eclipse 下载页面

可以选择 Eclipse IDE for Java Developers（85 MB）或者“Eclipse IDE for C/C++ Developers（68 MB）超链接。为了演示其安装插件的过程，本次选择下载 Java 版的 Eclipse，具体步骤如下。

（1）单击如图 17.1 所示的 Eclipse IDE for Java Developer 超链接。这时会进入下载的镜像

站点列表，可以选择一个最快的服务器进行下载。单击任何一个站点，就会开始下载。

注 意　经过实验，发现其实韩国（Korea）镜像站点的速度是最快的。

（2）下载后得到一个名为 Eclipse-java-ganymede-SR1-linux-gtk.tar.gz 的压缩文件。

（3）解压文件后得到一个名为 Eclipse 的目录，该目录中的 Eclipse 文件即为 Eclipse 的主文件，双击其就可打开 Eclipse 集成开发环境。

说 明　我们要安装的是 C/C++的开发环境，现在 Eclipse 中并无 C 开发环境，还需要下载一款名为 CDT 的插件，如果上一步选择的是 Eclipse IDE for C/C++的话，就不用下载 CDT 插件了。

（4）下载 CDT 可以在 Eclipse 网站上搜索 CDT，然后参照搜索的结果打开相关页面进行安装，也可以直接进入 http://download.Eclipse.org/tools/cdt/releases/ganymede/，选择适当的版本进行下载。下载时，同样会进入镜像站点列表，选择韩国站点进行下载即可。

（5）CDT 下载会得到一个名为 cdt-master-5.0.1.zip 的压缩文件，就是 Eclipse 的 C/C++插件。不用解压，直接在 Eclipse 中安装即可。

3．安装 CDT 插件

得到 CDT 插件后，需要进行安装。安装的具体步骤如下。

（1）运行 Eclipse，选择菜单“Help”|“Software Updates”命令，出现“Software Update and Add-ons”（软件更新与插件管理）对话框。

（2）单击“Available Software”（可用软件）标签，打开该标签页。

（3）单击“Add Site”（添加站点）按钮，出现如图 17.2 所示对话框。



图 17.2　添加 CDT 插件

（4）单击“Archive…”按钮，会出现选择文件对话框。定位到刚才下载的 CDT 插件即可将 CDT 加入到插件队列中。

（5）这时，会回到 Available Software 视图，显示所有的插件列表。选择刚才添加的 CDT 插件，单击 Install 按钮即自动安装该插件。

（6）安装结束后，重新启动 Eclipse 后就可进行 C/C++的开发了。

4．第一个 C/C++程序

在 Eclipse 创建一个 C 工程是件很容易的工作，只需要简单的几步。通过这几步创建的就是一个可实际运行的 C 工程框架。步骤如下。

（1）打开 Eclipse，选择 File|New|C Project 命令，开始创建第一个 C 工程。如图 17.3 所示。选择 C Project，意思是创建一个 C 的工程，然后单击“Next”按钮到下一步。

（2）出现如图 17.4 所示的对话框，在工程名称（Project name）中，输入要创建的 C 工程的名称，本例是 helloeclipse。

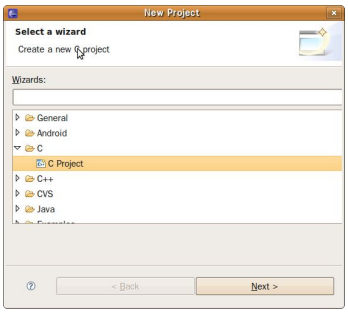


图 17.3　创建第一个 C 工程

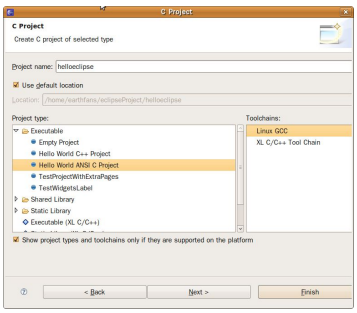


图 17.4　选择工程的相关信息

（3）在 Project type（工程类型）列表框中，选择适当的工程类型，本例选择了 Hello World ANSI C Project 类型。

（4）本例编译工具选择了 Linux GCC 选项。

（5）单击“Finish”按钮完成创建工作。



如果要设置作者信息、版权声明等内容，还可单击“Next”进行相关设置，如图 17.5 所示。然后一直单击“Next”按钮，直至最后出现“Finish”按钮，单击该按钮，则完成该项目的创建工作。

项目创建完成后，会出现如图 17.6 所示的界面。由于在创建时选择了 Hello World ANSI C Project 类型，所以创建的项目已经包含了一个可以正常运行，并且输出 Hello World 内容的主程序。现在，就可直接单击 Eclipse 中的 Run 按钮运行程序。该按钮会完成编译、连接的过程，然后自动运行代码，将输出显示在 Eclipse 的下方。这些均可从图 17.6 中看到。

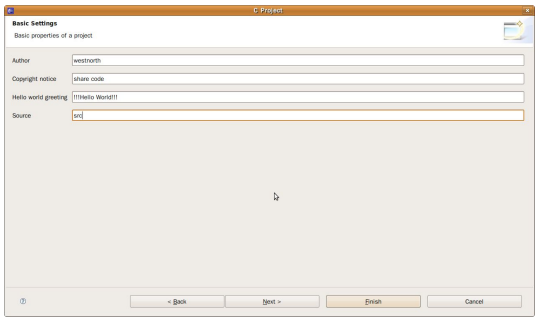


图 17.5　设置作者及版权信息

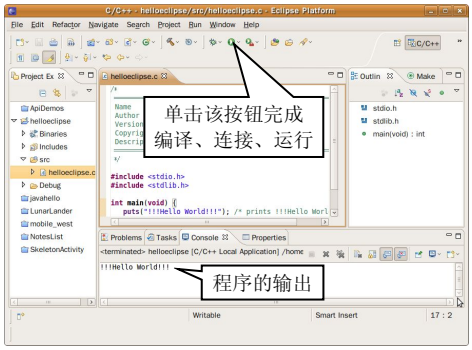


图 17.6　Eclipse 主界面

5．Eclipse 的其他应用

作为国际巨头的 IBM 公司主推的 IDE，Eclipse 的功能通过丰富的插件得到了极大的增强。而且也被越来越多地使用在其他各个领域。如互联网巨擘 Google 公司，也选择 Eclipse 作为其主推的手机开源操作系统 Android 的开发平台。下载了 Android 的 SDK 并且安装了 ADT 后，就可在 Eclipse 中进行 Android 平台软件的开发了。图 17.7 是 Android 模拟器的主界面。

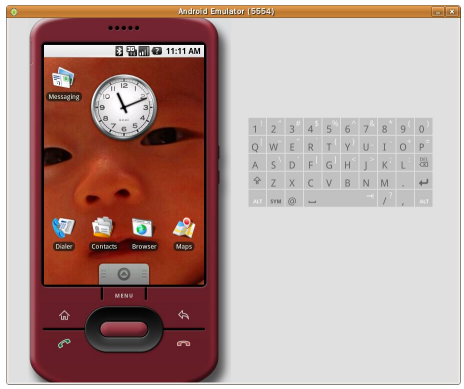


图 17.7　Android 模拟器主界面

图 17.8 是一个用 Eclipse 开发的 Android 游戏的界面截图。图 17.9 显示了调用 Android 的 API 可以完成的一些图形绘制工作。当然，这些演示还远不能体现 Android 的强大性能。有兴趣的读者可以去 http://code.google.com/android/查看详细资料。

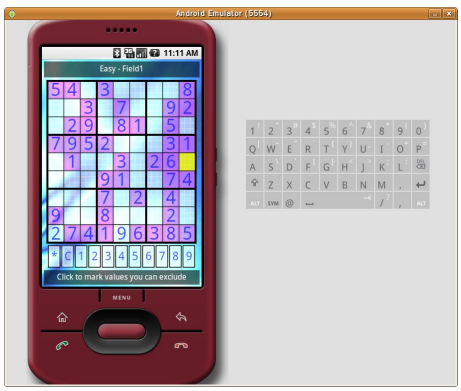


图 17.8　Android 游戏的界面截图

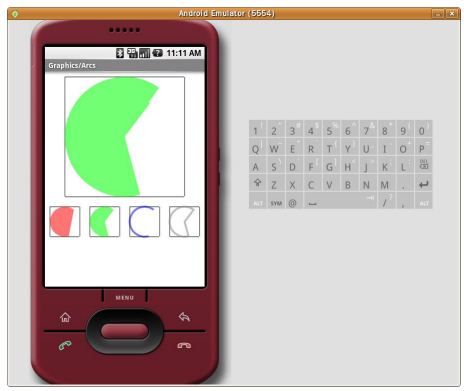


图 17.9　调用 Android API 绘制曲线

17.5　小结

本章介绍了 Linux 环境中进行 C/C++编程的一些基本知识，包括编译器 GCC 的使用、makefile 文件的编写，以及使用 GDB 调试程序的简单方法等。本章的重点内容是 GCC 与 GDB 的使用，这两个软件的使用其实并不难，多加练习掌握其基本使用还是比较容易的。本章的难点内容是 makefile 的编写。

第 18 章　Shell 编程

第 6 章介绍了 Shell 的一些基本命令，掌握了该章所介绍的命令后，就基本上可以完成系统的基本操作了。但有些时候，尤其是在一些不便于在终端输入命令或要执行大量命令的场合，就应该使用 Shell 编程来完成这项任务。Shell 编程可以使用所有的 Shell 命令。而且和大多数编程语言一样，有自己的语法结构、用户可以使用其变量定义、条件判断、循环等功能来完成各项复杂的任务。本章就简单地介绍一下 Shell 编程的基础知识。

18.1　Shell 脚本简介

本节简单地介绍脚本编程的相关基础知识，如脚本的特点、用途，以及特点等。脚本程序的基础知识包括脚本的结构与 Shell 的特殊字符。Shell 脚本的用户交互包括获得用户的输入、输出消息给用户，以及执行 Shell 的方法。

18.1.1　特点

Shell 脚本虽然也是一种编程语言。但与其他编程语言相比，有其独到的优点，但也有些不可避免的缺点。本小节就简单介绍了这些特点。

1．Shell 脚本的特点

Shell 编程对于 Linux 系统来说有着极其重要的意义。Linux 的启动就几乎完全依赖于系统 Shell 脚本的各种设定，而且系统中的许多命令其实也是一个个精心设计的脚本文件。由于 Linux 提供了许多参数复杂而又强大的命令，通过编写脚本程序可以很好地发挥这些程序的强大性能，来完成各种复杂的功能。

但脚本编程有缺陷，如执行效率低、没有图形界面、无法操作底层硬件设备等。所以 Shell 编程不太适合于下列场合。

- 要求执行效率高的场合，如大型数据处理等。Shell 文件是脚本语言，所以在执行效率上要比普通的高级语言如 C 等低得多。
- 数值计算：数字图像处理及大规模的数值处理。在这些情况下，应该选用 C/C++，Fortran 等。
- 需要图形界面的场合：如果用户需要有图形界面的程序，则 Shell 编程就无能为力了。这时，可以求助于 Python 等。
- 需要操作系统底层硬件的情况下：如果程序要对系统底层的硬件进行操作，Shell 并未提供相关硬件操作的功能，所以如果要直接操作系统底层硬件，就需要改用其他语言来编写程序。
- 需要跨平台执行的程序：由于 Shell 是严格的平台依赖性的语言，只能运行在 UNIX 及类 UNIX 平台上，如果需要良好的可移植性的程序，就不得不选择其他程序语言。

2．Shell 脚本的一般用途

使用 Shell 编程，可以极大程度地提高系统设置的方便程度。例如在“/etc/rcX.d”目录中系统启动时执行的配置文件均是脚本文件。里面的一个文件，如“/etc/rc2.d/S11klogd”部分内容如下：

```
#! /bin/sh
# /etc/init.d/klogd: start the kernel log daemon.
…
# Provides:             klogd
# Required-Start:      syslog
PATH=/bin:/usr/bin:/sbin:/usr/sbin
pidfile=/var/run/klogd/klogd.pid
case "$1" in
  start)
    log_begin_msg "Starting kernel log daemon..."
    # create klog-writeable pid and fifo directory
    mkdir -p /var/run/klogd
    chown klog:klog /var/run/klogd
    mkfifo -m 700 $kmsgpipe
    chown klog:klog $kmsgpipe
    …
exit 0
```

可以看到脚本文件以声明语句“#! /bin/sh”开头，以结束指令“exit 0”结束。除了声明语句外，其他以“#”开头的语句为注释语句。在脚本执行中，会忽略这些注释语句。但在脚本中增加注释语句是个好主意，因为这样大大提高了脚本文件的可读性。在脚本中使用了变量，也有判断语句，这些语句在接下来的篇幅中会加以介绍。

除了在系统设置中大量使用脚本文件外，系统提供的许多命令其实也是 Shell 脚本，如“/bin/gunzip”，其内容如下：

```
#!/bin/bash
PATH=${GZIP_BINDIR-'/bin'}:$PATH
exec gzip -d "$@"
```

可以看到，其实 gunzip 是调用了带有“-d”参数的 gzip 命令。

18.1.2 Shell 脚本程序的基本结构

要开始 Shell 编程，首先应该了解 Shell 脚本程序的基本结构。这样在实际编写 Shell 程序时才不会犯下低级错误。可以说，开始创建一个 Shell 脚本程序是极其简单的，只要记得遵循其结构即可。而其结构也很简单，只要在脚本的第一行说明脚本文件的适用 Shell 即可。Ubuntu 使用的 Shell 是 Bash，这也是类 UNIX 世界使用最为广泛的 Shell，所以脚本文件应该以如下语句作为开头。

```
#!/bin/bash
```

这句代码声明了解释该脚本的程序，该声明不是必须的。但其有两个作用，一是使用该声

明后，就可使用 bash 内建的指令，如变量声明、条件判断等高级特性，如果不使用该声明就只能使用命令的简单集合了；二是为了增强代码的健壮性，如果将没有声明的脚本移植到默认使用 Csh 的系统运行，系统会调用 Csh 来解释该脚本，这样可能会出现许多兼容性问题。

“#!”在 Shell 编程中被称为魔数。在魔数后应该指定一个运行该脚本的程序名的完整路径。必须注意的是，路径与程序名必须存在，否则运行脚本文件只会得到一个错误提示。由于脚本文件其实就是一些程序的自动运行，所以最简单的脚本就是罗列一些命令，每个命令占一行。在最后添加一句退出指令“exit”，这样就有了第一个简单的脚本文件。

```
#!/bin/bash
df >df_info
vi df_info
exit
```

使用任意文本编辑器，如 gedit，输入如上代码，并将其保存为 get_disk_space.sh。然后就可在终端中使用命令“get_disk_space.sh”来运行该脚本了。Bash 脚本文件应该以 sh 作为后缀。当然这并不是必须的，用户可以使用任意字符作为脚本文件的后缀。但出于兼容性方面的考虑，还是建议将脚本文件均以 sh 作为后缀。

18.1.3　Shell 脚本程序的几种特殊字符

在 Shell 中，有一些特殊字符，这些字符在脚本中代表着特殊的意义。如果将其当做普通字符来使用，可能会造成许多问题。所以将这些特殊字符列举如下，希望在使用时加以注意。

1．注释字符#

如果 Shell 中一行以#开始，则说明该行是注释行，用来解释该行或块的代码的功能，脚本运行时忽略这些行。如果有需要，#也可以放在一段语句的后面。注释的用法见下面的示例。

```
#注释行可以放在一行的开头
var=312                              #跟在语句后面的注释
```

2．双引号"

双引号可以强制 Shell 将一些特殊字符作为普通字符使用。这个看上去有点难以理解。由于命令行的许多行为与 Shell 脚本中是相同的。所以，我们分别运行带双引号与不带双引号的命令来观察其区别，以理解双引号的功能。示例如下：

```
westnorth@00:45 /bin$ cd /bin   #更改当前目录到/bin。
westnorth@00:45 /bin$ ls lo*    #执行不带双引号的命令，显示当前目录以 lo 开头的文件
loadkeys  login
westnorth@00:45 /bin$ ls "lo*"
#用双引号将命令括起来后，意义就变为：显示当前目录中的名字是“lo*”的文件
ls: cannot access lo*: No such file or directory
#显示的结果说明当前目录中没有名称是 lo*的程序
```

3．单引号'

与双引号类似，但其强制将所有特殊字符作为普通字符使用。下面介绍两个不同的命令，一个带双引号、一个带单引号，来观察其输出内容的区别。具体例子如下：

```
westnorth@00:48 ~$ echo "$BASH"   #使用带有双引号的命令来显示系统变量 BASH 的值
```

```
/bin/bash                                    #使用双引号时，$BASH 可以获得 Bash 的信息
westnorth@00:49 ～$ echo '$BASH'             #使用带有单引号的命令试图显示系统变量的值
$BASH
#如果使用单引号，则会将$BASH 原样输出，因为单引号取消了 $ 的特殊意义
```

4．转义字符

在一个特定的字符前加上\，意思是通知 Shell，该字符有了其他特殊意义。这些可以跟在\符号后面的特殊字符包括如下。

- \n：新行。
- \r：回车。
- \t：tab 键。
- \v：垂直 tab。
- \a：报警，也就是系统蜂鸣器响一声。

下面来看一些示例加深理解：

```
echo -e "\v\v\v"     #创建三个垂直 tab，注意这里的 echo 命令要带-e 参数
echo $'\n'               # 新行，如果不想带-e 参数，在变量前加 $ 也可实现类似功能
echo $'\a'               #报警
```

18.1.4 用户交互

对于一个编程语言来说，与用户交互是增强程序灵活性的一个重要手段。与用户交互是指显示提示信息给用户，并且可以得到用户输入信息来帮相应的处理。本节还介绍了运行 Shell 脚本代码的几种方式，这几种方式有其特定的功能，应该加以体会。

1．输出提示信息

在 Shell 中，可以使用 echo 命令将文本信息输出至终端。用法示例如下：

```
echo "hello world"
echo "$BASH"
#或者也可不用双引号，直接显示，这两种显示方式是等效的
echo hello world
echo $BASH
```

使用这种方式有个缺陷，就是一次只能显示一行内容。如果要显示多行文本，使用 echo 命令就显得不够优雅。

Shell 提供了一种更为方便的方法来显示多行文本，就是使用“<<”定界符。下面通过一个脚本代码来示例一下“<<”定界符的使用。

```
#!/bin/bash
cat <<EOF   #在“<<”操作符后跟的“EOF”是定界符，Shell 将一直输出后面的文本，直到遇到“EOF”，定界符 EOF 可以是任意字符
hello linux!
hello ubuntu!
goodbye windows!
welcome to real world!
EOF                                  #遇到”EOF”，输出结束
```

```
exit
```

输出应该类似这样：

```
hello linux!
hello ubuntu!
goodbye windows!
welcome to real world!
```

2．获得用户输入

一个完整的程序设计语言应该具备与用户交互能力。所谓交互，就是程序应该可以对用户的输入产生反应。Shell 脚本获得用户输入的函数是 read。其用法见下面的脚本：

```
echo "please input a number: "
read user_in                          #读取用户输入，并存入名为 user_in 的变量中
echo "the number of you input is $user_in"
```

其输出类似这样：

```
please input a number:
4829                                  #该值为用户输入的值
the number of you input is 4829       #在输出中引用了保存输入值的变量
```

3．运行脚本的方法

一个脚本文件，如本例的 get_disk_space.sh，在编写完成后，使用 ls -l 命令查看文件详情，应该如下：

```
westnorth@16:02 ～/Shell$ ls -l
总用量 4
-rw-r--r-- 1 westnorth westnorth 41 2008-11-23 16:02 get_disk_space.sh
```

参照 6.4.1 节相关内容理解输出内容的含义，就可以知道现在该文件只有读权限，没有运行的权限，所以还无法运行。要给其添加运行权限，可以使用 chmod 命令，代码如下：

```
chmod u+x get_disk_space.sh
```

现在再用 ls -l 查看文件详情，可以看到该文件已经拥有了运行权限，在命令行路由输入如下内容，运行第一个脚本文件。

```
./get_disk_space.sh
```

出于安全方面的考虑，类 UNIX 系统并未将当前目录列入可执行程序的搜索路径路由，意思是如果可执行文件放在当前目录，当用户试图直接输入该程序名想使其运行时，会得到错误提示，如下所示：

```
westnorth@16:17 ～/Shell$ get_disk_space.sh
bash: get_disk_space.sh：找不到命令
```

所以，记得在运行当前目录的可执行程序时，要显示指定其位置，如“./get_disk_space.sh”，这样程序才会正常运行。运行程序还有另外一种方法，就是显式指定方式。这种方法与前一种方法基本类似，不过不需要脚本文件有运行的权限也可运行该脚本。可用如下所示命令来显式

运行 get _disk_space.sh 脚本。

```
bash get_disk_space.sh
```

18.2 变量与数组

像其他程序语言一样，Shell 语言支持变量赋值操作。Shell 的变量声明时没有类型的概念。而且从某种意义上来说，Shell 变量不需要声明，在给变量赋值的同时就指定了变量的类型及变量的值。变量的使用极大地增强了脚本的灵活性。

1. 变量声明

变量的声明很简单，格式是“变量＝值”，注意在等式的两边不能有空格。如果两边均有空格，像这样“变量 ＝ 值”则是测试变量与值是否相等。而如果只在等号前面有空格，如“变量 ＝值”，Shell 会寻找以变量为名的命令，然后加“＝值”参数运行。正确地使用变量，应该像下面这样。

```
line_num=35                #声明一个数值型变量，值是 35
name='westnorth'           #声明一个字符串型变量，内容是 westnorth
```

除了这样声明变量外，还可以声明一个数值型变量为二进制、八进制或十六进制类型。定义的方法如下：

```
let "num_dec = 40"
echo " the num is = $num_dec"
#输出结果为“40”，这是最普通的定义数值的方法，默认是十进制数值
let "num_oct = 040"
echo "oct= $num_oct"
#显示结果为“32”，这是定义八进制数值的方法，只需在数字前加上数字“0”即可
let "hex = 0x40"
echo "hex= $hex"
#显示结果是“64”，这是定义十六进制数值的方法。在变量的值前加上“0x”或“0X”即可
```

2. 给变量赋一个数学表达式的值

如果要给变量赋一个数学表达式的值，就不能直接使用等号了。使用“a=3+2”，并不会让 a 等于 5，而是让 a 等于“3+2”，成为一个字符串。正确的方式应该是使用 set 关键字，示例如下：

```
set a=3＋2
set a1=3*3-$a
```

这样使用才可以让 a 等于 5，然后让 a1 等于 4。

3. 给变量赋一个命令的运行结果

还可以给变量赋予一个命令的运行结果，如：

```
b=$(ls -l)
```

这样，就将“ls -l”的运行结果赋予变量 b。只要在标准输出中有输出内容的命令均可将其输出内容赋予一个变量。这样就从某种程度上提高了使用效率。因为系统提供了许多极强大的对于文本、数据进行操作的命令，如果能直接将其输出保存到一个变量中，会给使用带来极大

的便利。

4．使用变量

要在 Shell 中提取变量，应该使用操作符“$”。命令行中声明变量，然后引用变量的用法示例如下：

（1）声明变量，使用如下命令：

```
westnorth@08:10 ～ $ name='westnorth'    #声明变量
```

（2）引用变量，使用如下命令：

```
westnorth@08:10 ～ $ echo $name          #引用变量，使用$
westnorth                                #输出变量定义时赋予的值
```

在前文曾经提及，也可将命令的输出赋予变量。用法示例如下。

（1）将 ls -l 命令的输出赋予变量 content，使用如下命令：

```
westnorth@08:10 ～ $ content=$(ls -l)
```

（2）引用变量 content，将其显示出来，使用如下命令：

```
westnorth@08:11 ～ $ echo $content
#使用$content 变量引用变量，注意这种方式引用变量，会消除所有回车标记
#下面是命令的输出
total 52160 -rw-r--r-- 1 westnorth westnorth 11000809 2008-06-16 15:09
Addison Wesley Advanced Programming In The Unix Environment (2005).2Ed.pdf
-rw-r--r-- 1 westnorth westnorth 850924 2008-11-24 07:49 #AdvancedBashScript
_Guide_cn.txt#  -rw-r--r--  1  westnorth  westnorth  756816  2006-05-15  23:03
AdvancedBashScript_Guide_cn.txt  -rw-------  1  westnorth  westnorth  3866012
2006-02-13 18:38
```

（3）由于前面的输出不太理想，换种方式，将引用变量的代码用双引号括起来。使用如下命令：

```
westnorth@08:12 ～ $ echo "$content"
#使用带双引号的$content 引用变量，这种方式输出的内容保留了原有回车符
total 52160
-rw-r--r-- 1 westnorth westnorth 11000809 2008-06-16 15:09 Addison Wesley
Advanced Programming In The Unix Environment (2005).2Ed.pdf
-rw-r--r-- 1 westnorth westnorth  850924 2008-11-24 07:49 #AdvancedBash
Script_Guide_cn.txt#
-rw-r--r-- 1 westnorth westnorth  756816 2006-05-15 23:03 AdvancedBash
Script_Guide_cn.txt
-rw------- 1 westnorth westnorth  3866012 2006-02-13 18:38 Advanced Linux
Programming.pdf
westnorth@08:12 ～/linux/program/system$ echo '$content'
#如果使用单引号试图引用变量，则会得到如下结果
$content
```

但有时候会出现普通文本与命令输出混合显示的情况，这种情况下输出命令有点特殊，下

面通过一个示例进行介绍。具体步骤如下。

（1）定义一个变量，作为显示之用。使用命令如下：

```
westnorth@10:13 ～/Shell$ num=18                                    #定义变量
```

（2）使用前文所述方法引用变量，使用命令如下：

```
westnorth@10:16 ～/Shell$ echo $numyears can smoke
#如果要将变量值与普通文本混合显示，直接使用会出现错误，上述代码会指示 Shell 去寻找一个名叫 numyears 的变量，如果用户并未定义该变量的话，显示就如下所示
can smoke
```

（3）使用正确的方法显示文本与命令混合的情况，命令如下：

```
westnorth@10:16 ～/Shell$ echo ${num}years can smoke
#使用大括号将变量括起来，就可以混合显示变量与普通文本
18years can smoke
```

上文中使用的“echo”命令是回显指令，显示后面跟着的内容。

5．取消变量

如果想从脚本中取消一个已定义的变量，可以使用 unset 命令。下面通过一系列代码来理解该命令的用法。具体步骤如下。

（1）声明一个变量，使用如下命令：

```
westnorth@09:27 ～$ name='westnorth'     #声明一个变量,名为 name,值为 westnorth
```

（2）引用该变量值，将其显示出来。使用命令如下：

```
westnorth@09:29 ～$ echo "name=$name"   #显示该变量值
name=westnorth                          #显示了变量的值
```

（3）现在取消该变量，使用命令如下：

```
westnorth@09:29 ～$ unset name
#取消该变量
```

（4）现在显示该变量，注意查看输出。使用命令如下：

```
westnorth@09:31 ～$ echo "name=$name"
name=                                   #显示该变量会得到一个空的输出结果
```

6．系统变量

系统中定义了许多变量，有的变量可以获得参数，有的变量可以定义 Bash 的行为，可直接使用这些变量来提高效率。

参数位置变量：“$”后跟一个数字可以获得脚本文件运行时输入的参数值。“$0”是当前脚本的文件名。然后“$1”是第一个参数，“$2”是第二个参数，依次类推。如果参数值多于 9 个，使用时需要加上大括号，如用${10}来获得第 10 个参数。来看如下的例子：

```
westnorth@11:09 ～/Shell$ ./get_var.sh do we have some wrong
#运行该脚本文件。注意在运行时要给该脚本赋予多于 3 个的参数，才能得到正确的输出，下面是输出结果
the number 3 argument  is have
```

```
#get_var.sh 是个简单的显示参数的脚本，内容如下所示
#!/bin/bash
echo the number 3 argument  is "$3"
echo
exit
```

其他系统变量如下所示。

- $#：执行脚本文件时输入参数的个数。如在 get_var.sh 中加入一行：

```
echo the total argument is "$#" argument
```

就会得到当前脚本输入参数的个数。

- $*：将所有的参数作为一个单词来引用。
- $@：与“$*”有相同的作用，但将每个参数作为一个单词来引用。这就意味着得到的引用是一个字符数组。

7．内部变量

下面列出一些常用的内部变量，可以在脚本中引用这些变量，也可以在终端中使用“echo $变量名”来查看其值。如“echo $BASH”就可以查看 Bash 程序的位置。而且要注意的是，用户定义的变量不能与这些内部变量同名。下面所列出的是一些 Shell 内部变量。

- $?：得到命令、函数或者脚本的退出状态。
- $$：当前运行的脚本文件的 PID 号。
- $BASH：Bash 程序的位置。
- $BASH_VERSINFO[n]：Bash 程序的版本信息，如 n=0，则得到主版本号，如 n＝5，则会得到 Bash 的架构信息。
- $FUNCNAME：当前函数名。
- $GLOBIGNORE：　一个文件名的模式匹配列表，如果在 file globbing 中匹配到的文件包含这个列表中的某个文件，那么这个文件将被从匹配到的文件中去掉.
- $GROUPS：当前用户所属的组。这是一个数组文件，包含了当前用户所属所有的组。可以用“echo ${GROUPS[1]}”来得到其值。
- $HOME：用户的主目录。
- $EDITOR：脚本默认的编辑器，一般为 VI 或 Emacs。
- $HOSTNAME：显示当前机器的名称。
- $HOSTTYPE：显示当前计算机的架构类型，如 i486、i686。
- $MACHTYPE：显示系统的硬件相关信息，一个可能的结果是 i486-pc-linux-gnu。
- $IFS：指定 Shell 解析字符串时分隔单词的分隔符，默认是空格、TAB、回车等，可以修改其值。如字符串各单词间是用逗号分隔的，就可以使用“IFS=’,’”指定将逗号作为新的单词分隔符。
- $LINENO：得到当前行的行号。
- $OSTYPE：操作系统类型，一个可能的输出是 Linux-gnu。
- $PATH：指定可执行程序的路径。在脚本中引用外部命令时，就在这些路径中寻找可执行程序。其内容可能是“/home/westnorth/bin:/usr/local/sbin：/usr/local/bin: /usr/sbin: /usr/bin:/sbin: /bin: /usr/game”。PATH 中可能有多个目录，各目录间用“:”隔开。

8．数组

在 Bash 中可以像其他编程语言一样，使用数组进行操作。Bash 中数组中元素同样也没有数据类型的概念，而且 Bash 中只能使用一维数组。声明一个数组有两种方法。

```
declare -a array                    #第一种方法
array[23]=“BMW”                     #第二种方法
```

而其使用则很简单，像使用 Bash 中其他变量一样，使用“$”提取其值。不过，在数组外要加大括号。如：“${array[23]”。下面是一个定义数组，并提取数组中元素的脚本文件示例。

```
#!/bin/bash
declare -a arr1                     #显式声明一个数组
arr1[1]="abcd"                      #给该数组的第 1 个元素赋初值为“abcd”
array[11]=11;                       #直接定义一个数组，并同时将第 11 个元素赋值为 11.
echo "array No.11 is ${array[11]}" #显示两个数组的元素
echo "arr1 No.11 is ${arr1[1]}"
echo "unknow arr1 is ${arr[2]}"     #显示未赋值的元素
echo "unknow array is ${array[2]}"
```

该脚本的输出如下所示：

```
array No.11 is 11
arr1 No.11 is abcd
unknow arr1 is
unknow array is
```

可以看到，如果提取的是已赋值的元素，则显示该元素的值。否则会显示一个空字符串。

18.3 算术操作符

Shell 支持算术运算符，这为对变量进行简单的数值运算提供了方便。只要所操作的常量或变量是数值，就可以使用下列算术操作符：

- +：加法运算；
- -：减法运算；
- *：乘法运算；
- /：除法运算；
- **：幂运算运算；
- %：取模运算。

下面以一个脚本来示例这些运算符的使用：

```
#!/bin/bash
#算术运算符的示例脚本
let num1=30
let num2=4
echo "num1=${num1} num2=${num2}"
let total_num=${num1}+${num2}
```

```
echo "$num1+$num2=$total_num"           #加法运算
let multi_num=${num1}**${num2}
echo "$num1^$num2=$multi_num"           #幂运算
let mod_num=${num1}%${num2}
echo "${num1} mod ${num2} = $mod_num"   #取模运算
${num1}+=3
echo $num1
```

该脚本的输出应该类似这样：

```
num1=30 num2=4
30+4=34
30^4=810000
30 mod 4 = 2
```

熟悉 C/C++的用户对于自操作形式的表达式应该很熟悉，比如“+=”就是将自身加上一个数，而“-=”就是将自身减去一个数。Shell 支持的自操作运算符有如下几种：

- +=：加等于；
- -=：减等于；
- *=：乘等于；
- /=：除等于；
- %=：取模赋值。

还是以一个脚本为例说明其功能，脚本内容如下：

```
#!/bin/bash
num1=46
num2=7
echo "num1=$num1 num2=$num2"
let "num1 += 18"                        #让 num1 加上 18
echo "num1+=18 result: $num1"
echo "now num1=$num1"
let "num1 /= num2"                      #让 num1 除以 num2
echo "num1 /= num2 result: $num1"       #显示 num1 的值
exit
```

其输出结果如下：

```
num1=46 num2=7
num1+=18 result: 64
now num1=64
num1 /= num2 result: 9
```

18.4　条件判断语句

在脚本中，很多时候需要根据不同的情况执行不同的操作。这样，就需要使用条件判断语句。常用的条件判断语句是 if 语句、case 语句。下面分别予以介绍。

18.4.1 if 语句

if 是一个比较常用的 Shell 指令，用来测试一个条件。如果条件满足，则执行一些指令；如果条件不满足，则执行另一种代码。类似于“如果明天天晴，那么就去春游”之类。所以其原型在现实生活中可以找到许多对应的参照物，还是比较容易理解的。

1．if 的语法

该命令的具体语法如下：

```
if 条件 1; then
动作 1
动作 2
elif 条件 2; then
动作 3
动作 4
else
动作 5
     动作 6
fi
```

其意义是，如果满足“条件 1”，则执行“动作 1、2”；否则，如果满足“条件 2”，则执行“动作 3、4”；如果“条件 1”与“条件 2”均不满足，就执行“动作 5、6”；fi 是 if 命令结束的标记。在实际使用中，不一定要使用这个完整形式，if 命令的最简格式是：

```
if 条件 1 ; then
  动作 1
fi
```

意思是满足“条件 1”的情况下，执行“动作 1”。不过对于这种只有一个条件的判断，一个更好的解决方案是“[条件 1]&&动作 1”。这样也可以在满足“条件 1”的情况下执行“动作 1”。为了更清楚地理解这个条件判断语句，还是以一个脚本来说明，脚本文件的代码如下：

```
#!/bin/bash
year=2008
 if [ "$year > 2000" ]; then                                    #使用 if 进行测试
    echo "this year is 21th "
 else
    echo "this year is not in 21th"
fi
 [ "$year > 2000" ] && echo "this year is 21th by second way"  #第二种测试方法
exit
```

脚本的输出如下：

```
this year is 21th
this year is 21th by second way
```

2．使用逻辑判断语句

有时需要判断比较复杂的条件情况，如有时需同时满足几个条件才可执行某项动作；而有时只需要满足几个条件中的任何一个就可执行某项动作。在这两种情况下，就可分别使用逻辑与操作符“&&”、逻辑或操作符“||”。下面给出一个脚本，来介绍这两种逻辑操作符的使用方法。

```
#!/bin/bash
#逻辑“与”、逻辑“或”操作行的示例代码
num1=35
num2=58
num3=87
num4=12
if [ "$num1" \< "$num2" ] && [ "$num2" \< "$num3" ]    #测试是否同时满足这两个条件
#注意，在使用比较操作符“<”时，需使用转义字符“\”
then echo "they are sorted"
fi
if [ "$num1" \> 50 ] || [ "$num2" \> 50 ]  #测试是否满足这两个条件中的任何一个
#同样，在使用比较操作符“>”时，需使用转义字符“\”
then echo "there is one num bigger then 50 at least"
fi
exit
```

该脚本的输出类似这样：

```
they are sorted
there is one num bigger then 50 at least
```

18.4.2　case 语句

有时需要判断的条件过多，使用 if 语句会导致多层嵌套，既容易出错，又不直观。Shell 提供了一种更好的方法来适应这种情况，就是 case 语句，其格式是：

```
case "变量" in
  变量 a|变量 b|变量 c)      #如果变量与三者中任意一个变量相等，就执行下面的“语句 1”
     语句 1
      ;; #这条判断分支语句结束
  变量 d)                   #如果等于“变量 d”，则执行“语句 2”
     语句 2
     ;;
   *)                       #前面条件均不满足时执行“语句 3”，
语句 3
     ;;
esac                        #case 语句结束
```

下面以“/etc/rc2.d/S10acpid”脚本文件中部分语句为例介绍 case 的用法。

```
case "$1" in                #“$1”代表第一个参数
  start)                    #如果第一个参数是“start”，则执行语句直到“;;”。
```

```
        [ -f /proc/modules ] && load_modules
        log_begin_msg "Starting ACPI services..."
        start-stop-daemon --start --quiet --oknodo --exec /usr/sbin/acpid -- -c
/etc/acpi/events $OPTIONS
        log_end_msg $?
        ;;                      #第一个分支语句结束
      stop)
        log_begin_msg "Stopping ACPI services..."
        start-stop-daemon --stop --quiet --oknodo --retry 2 --exec /usr/sbin/
acpid
        log_end_msg $?
        ;;                      #第二个分支语句结束
      restart)
        $0 stop
        sleep 1
        $0 start
        ;;                      #第三个分支语句结束
      reload|force-reload)  #如果第一个参数是“reload”或“force-reload”，则执行下
列语句
                              #这也是常用的复合判断的一个技巧
        log_begin_msg "Reloading ACPI services..."
        start-stop-daemon --stop --signal 1 --exec /usr/sbin/acpid
        log_end_msg $?
        ;;
      *)                      #如果第一个参数不符合上面任一选项，则执行下列语句
        log_success_msg "Usage: /etc/init.d/acpid {start|stop|restart|reload|
force-reload}"
        exit 1
    esac
```

18.4.3 select 语句

select 也是判断语句的一种类型，其作用是在终端上显示一个菜单。用户可以从中选择其中一个选项，那么脚本就会执行与该选项对应的操作。其语法结构是：

```
select 变量 in 列表
do
命令...
break
done
```

在产生菜单时，会自动给列表中的字符按照顺序编号。需要注意的是 select 语句在显示提示信息时，使用的是系统变量 PS3，所以一般在使用 select 命令前，会修改 PS3，以适应自己的需要。示例代码如下：

```
#!/bin/bash
PS3='Who am I: '            # 修改提示信息.
```

```
echo
select iam in "cat" "dog" "carrot" "tiger" "panda"
do
echo
echo "Your think I am  $iam."
echo -n "But ......"
sleep 1
echo "..."
if [ "$iam" = panda ]; then
    echo "you are right"
    break                    #如果选择了 panda 项，则退出 select 命令
else
    echo "your are wrong"    #否则，仍然显示提示信息，提示用户输入
fi
echo
done
exit 0
```

脚本的输出应该类似于这样：

```
1) cat     3) carrot  5) panda
2) dog     4) tiger           #这就是 select 命令自动编号的输出结果
Who am I: 1                   #输入 1
Your think I am  cat.
But .........
your are wrong                #输入错误，重新回到提示状态
Who am I: 5                   #输入 5
Your think I am  panda.
But .........
you are right                 #输入正确，退出脚本
```

18.4.4　各种测试操作

Shell 支持许多的测试操作，可以用在许多不同的场合，然后根据测试的结果，脚本可以执行完全不同的操作，以增强代码的灵活性与安全性。现将这些 Shell 支持的测试操作列于下，以便查阅（filename 是用户指定的文件名）。这里并没有给出示例代码，这些命令的示例可以参照相关脚本进行理解。

1．文件测试操作

文件操作符是用于测试文件属性的命令。可以测试文件几乎所有的属性，如大小、创建时间、文件长度、文件类型等。具体如下所示。

- -e filename：文件存在则返回真。
- -f filename：文件是普通文件（不是目录或者设备文件）则返回真。
- -s filename：文件长度不为 0 则返回真。
- -d filename：文件是个目录则返回真。
- -b filename：文件是个块设备（软盘，cdrom 等等）则返回真。

- -p filename：文件是个管道则返回真。
- -h filename：文件是个符号链接则返回真。
- -r filename：当前用户对该文件具有读权限则返回真。
- -w filename：当前用户对该文件具有写权限则返回真。
- -x filename：当前用户对该文件具有运行权限则返回真。
- -O filename：当前用户是文件的所有者则返回真。
- filename1 -nt filename2：文件 filename1 比 filename2 新则返回真。
- filename1 -ot filename2：文件 filename1 比 filename2 老则返回真。

2．数值比较操作

下面的操作符应该像普通的运算符一样使用，也就是说在其前后各有一个变量或常量。

- -eq：等于，用法示例“if ["$num1" -eq "$num2"]”，下同。
- -ne：不等于。
- -gt：大于。
- -ge：大于等于。
- -lt：小于。
- -le：小于等于。
- <：小于（需要将其放置于双括号中）。
- <=：小于等于（需要将其放置于双括号中）。
- >：大于（需要将其放置于双括号中）。
- >=：大于等于（需要将其放置于双括号中）。

3．字符串比较操作

字符串操作在实际脚本中得到了极其广泛的使用。如用来测试文件名、测试文件的一部分文本内容等。具体的命令如下。

- =：等于，用法示例“if ["$num1" = "$num2"]”。
- ==：与“=”相同。
- !=：不等于。
- <：小于，在“[]”中，需要在其前加“\”进行转义。而在“[[]]”结构中，可直接使用。用法示例“if [["$var1" < "$var2"]] ”或 “if ["$var1" \< "$var2"]”。
- >：大于，与小于号的使用方法相同，在“[]”结构中需进行转义。
- -z：字符串为空。
- -n：字符串不为空。

18.5 循环语句

在各种编程语言中，循环语句也是极其重要的一个部分。使用循环语句可以快速多次执行某些具有共同特点的语句，因此被广泛运用于许多重复性的操作过程中。Shell 中循环控制语句有 for、while、until 与 repeat 语句几种。

18.5.1 for 语句

for 语句对于稍有编程经验的用户来说都不陌生。其主要功能就是测试变量是否在一个列表中，如果在列表中，就执行相应的操作。但 Shell 编程中列表必须在 for 语句头中显式指定，

所以灵活性不如其他高级语言，其语法结构如下：

```
for 变量 in [列表]
do
          命令集
done
```

其示例代码如下：

```
#!/bin/bash
for nation in "China" "U.S.A" "U.K" "Swiss" "Russia"   #指定了一个列表
do
  if [ "$nation" == "China"  ];then                    #测试值是否在列表中
      echo "Our Country is $nation"
      continue
  else
      echo "and there are other country in earth,like  $nation"
  fi
done
exit
```

输出结果应该类似这样：

```
Our Country is China
and there are other country in earth,like  U.S.A
and there are other country in earth,like  U.K
and there are other country in earth,like  Swiss
and there are other country in earth,like  Russia
```

for 语句的“列表”既可以像前例一样使用一些字符的列表，也可以使用提前定义的数组。看下面的例子：

```
#!/bin/bash
exefile="/bin/cat
 /bin/ed
 /bin/echo
 /bin/myfile
 /bin/loadkeys"
#定义一个用于操作的数组
for myfile in $exefile
do
  if [ ! -e "$myfile" ]       #文件不存在吗
  then
    echo "sorry,$myfile is not exist no longer"
    continue         #继续下一步
 else
  echo "oh,the file $myfile exist"
 fi
done
```

```
exit 0
```

脚本的输出如下：

```
oh,the file /bin/cat exist
oh,the file /bin/ed exist
oh,the file /bin/echo exist
sorry,/bin/myfile is not exist no longer
oh,the file /bin/loadkeys exist
```

18.5.2 while 语句

while 结构用于判断一个条件，如果该条件满足，就一直进入循环。条件满足，才退出循环。这个结构适于一些循环次数未知的情况。while 的结构如下：

```
while [条件]
do
命令...
done
```

下面一个脚本用来显示当前目录中的所有文件，并将其编号输出。

```
#!/bin/bash
cur_file="$(ls)"                        #获得当前目录中所有文件列表,并存入cur_file
cur_file_num="$(ls |wc -l)"             #获得当前目录中总文件数
num=0                                   #设定开始显示的文件编号
while [ "$num" -le "$cur_file_num" ]    #如果文件起始编号小于总文件数
do
   for filein in $cur_file;do
          echo "the No.${num} file is $filein"
        let "num+=1"
   done
done
var=""
echo "do you undesrstand this script?"
echo if you do,type Isee to quit
while [ "$var" != Isee ]    #如果 var 不为“Isee”则执行下列代码，否则结束循环
do
      read var                  #读取用户输入，存入变量 var 中
      if [ "$var" != Isee ]     #判断用户输入，如果不为“Isee”
            then
                  echo "you input is $var"
             else               #如果用户输入”Isee”
                  echo "quit"
      fi
done
exit
```

脚本的输出类似这样：

```
the No.0 file is bit_opt.sh
the No.1 file is bit_opt.sh~
the No.2 file is df_info
the No.3 file is get_disk_space.sh
the No.4 file is get_var.sh
the No.5 file is get_var.sh~
…
the No.57 file is var_test.sh~
do you undesrstand this script?
if you do,type Isee to quit
efe
you input is efe
hello world
you input is hello world
Isee
quit
```

18.5.3　until 语句

until 结构与 while 类似，也用于判断一个条件，然后执行相应的操作。但其与 while 不同的是，在条件不满足的条件下才会执行相应的操作。until 的结构如下：

```
until [条件]
do
            命令
done
```

如果用 until 命令来改写前面使用 while 命令的脚本，则其中相关的内容可改为：

```
until [ "$var" = Isee ]     #原代码是“while [ "$var" != Isee ]”，可参照对比一下其不同之处
do
       read var
       if [ "$var" != Isee ]
       then
           echo "you input is $var"
       else
           echo "quit"
       fi
done
```

18.5.4　循环语句的一些控制语句

循环语句的一般结构就是前面介绍的三种：for、while、until。这些语句的组合可以完成一系列复杂的功能。为了更好地控制程序的流程，Shell 还提供了一些控制这些循环的命令，如 break、continue 等。在了解这些控制命令之前，先来看一下执行一语句的复合结构。前述的循环语句均是可以进行多层嵌套的。如：

```
for var in ; do                                    #第一层循环
    in_var=1
    while [ $in_var -le max_num ]; do              #第二层循环
          until [ somethin ]; do                   #第三层循环
…
         done                                      #第三层循环结束
    in_var=$(( in_var + 1 ))
   done                                            #第二层循环结束
done                                               #第一层循环结束
```

以一个经典入门性的绘制星号三角形的代码来演示循环嵌套功能，代码如下：

```
#!/bin/bash
times=0
time2=0
while [ $times -lt 8 ]; do          #设置三角形的高度，也就是说三角形有几行高
     while [ $time2 -lt $times ]; do    #三角形的第几行则绘制几个星号，如第 3 行
                                        就绘制三个星号
           echo  -n '*' #使用“－n”可以阻止 echo 产生新行，这行的功能是在当前行
                        上绘制星号
           let "time2+=1"
     done
let time2=0                        #将行上的星号计数器重新置 0
let "times+=1"
echo #换行
done
exit
```

输出是一个三角形：

```
*
**
***
****
*****
******
*******
```

在循环语句执行过程中，有时需要一些特殊的语句，如退出循环、不执行本循环中其他语句直接进入到下一个循环。执行这两项功能的语句分别是 break、continue。

1．break

break 命令的作用是退出当前循环。也就是说从循环语句中跳出，执行循环语句外的其他语句。

2．continue

continue 命令的作用是不再执行本循环语句中其余的语句，但并不退出当前循环。下面是一个脚本文件，分别显示了 break 语句与 continue 语句的不同之处：

```
#!/bin/bash
num1=0
while [ "$num1" -lt 5 ]; do     #while 结构开始
    echo this is $num1
    if [ "$num1" -eq 2 ]; then #第一个 if 语句，判断 num1 是否等于 2
      echo "use continue"
      let "num1+=1"
      continue
#如果 num1 等于 2，使用 continue 则不再执行 if 语句外的其他命令，包括“语句 1”与“语句 2”等。，而是执行下一步循环，即 num1 等于 3 的循环
    Fi                          #第一个 if 语句结束
    if [ "$num1" -eq 3 ]; then  #第二个 if 语句，判断 num1 是否等于 3
            echo "use break,current num is $num1"
            break               #
    fi                          #第二个语句结束
echo "not use continue"         #语句 1
echo "not use break"            #语句 2
let "num1+=1"
done                            #while 结构结束
exit
```

脚本的输出结果如下：

```
this is 0
not use continue
not use break
this is 1
not use continue
not use break                   #num1 等于 1、2 时，输出并没有什么特别
this is 2                       #num1 等于 2 时，执行了第一个 if 语句
use continue                    #使用 continue 后，没有“not use…”的输出
this is 3                       #num1 等于 3 时，执行了第二个 if 语句
use break,current num is 3
#使用 break 语句，虽然不符合退出的条件(num =5)也会强制退出循环。所以导致了后面并没有输出 num1 大于 3 时的任何内容
```

18.6　使用函数

像其他编程语言一样，Shell 提供了函数功能。不过，这个函数功能比较简单，和其他高级语言提供的函数功能相比，功能上限制很多，但用于执行某些重复性的工作，还是具有很强实用价值的。Shell 中使用函数很简单，先声明函数，然后就可直接使用函数名来调用函数了。函数的声明有两种方式。

- 函数声明方法 1，语法如下：

```
function 函数名 {
```

```
命令...
}
```

- 函数声明方法 2，语法如下：

```
函数名 () {
命令...
}
```

使用函数就更简单了，直接在脚本中使用函数名即可。下面以一个例子来说明函数的使用方法：

```
#!/bin/bash
#opt_func.sh
get_bigger () {   #声明了一个名为 get_bigger 的函数，Shell 的函数必须在使用前声明。
    if [ $1 -gt $2 ]; then      #Shell 函数要想读取用户传入的参数，可以使用读取 $
                                1、$2 的方法
            return 111    #返回值是 111，这个值可以在外部使用“$?”来调用
    else
            return 222
    fi
    exit
}
echo input number 1:
read var1
echo input number 2:
read var2
get_bigger "$var1" "$var2"  #调用函数 get_bigger，后面跟的是要传入函数的两个参数
if [ "$?" -eq 111 ]; then   #“$? ”是函数的返回值，可以通过读取返回值来判断相关
                            状态信息
   echo "the number 1 is bigger"
else
   echo "the number 2 is bigger"
fi
exit
```

脚本的输出类似于这样：

```
input number 1:
283
input number 2:
94
the number 1 is bigger
```

18.7 脚本调试

如果有其他语言的编程经验，会发现 Shell 编程语言似乎并没有提供方便的调试功能，这样做一个显而易见的理由是：Shell 脚本均由纯文本文件组成，容易检查其中的错误。而且使用 echo 命令可以很方便地显示输出。但是，为了发现 Shell 中的错误，还是需要一个稍微强大些

的调试辅助工具。



在 Shell 感觉到容易出错的地方，多多使用 echo，这样可以为查明问题提供很大的帮助。

虽然 Shell 并未提供强大的调试工具，但还是可以通过一些方法来获得脚本的语法错误信息与得到动态执行结果。就是通过使用 bash 的相关命令行选项。

- bash -n scriptname：检查脚本的语法错误，注意其只是检查脚本的语法错误，而对于语义错误是不能检查的，在脚本中插入 set -n 或 set -o noexec 也有类似的效果。
- bash -v scriptname：在执行每个命令前先输出这个命令。在脚本中插入 set -v 或 set -o verbose.也可达到相同的效果。通常是将-n-v 合并在一起使用。
- bash -x scriptname：输出每个命令的运行结果。在脚本中插入 set -x 或 set -o xtrace 也可。

下面是一个使用 x 参数运行脚本的结果：

```
westnorth@01:36 ~$ bash -x Shell/opt_func.sh
+ echo input number 1:
input number 1:
+ read var1
123
+ echo input number 2:
input number 2:
+ read var2
456
+ get_bigger 123 456
+ '[' 123 -gt 456 ']'              #这时已经来到了函数 get_bigger 境内，开始比较
+ return 222                       #函数的返回值 222
+ '[' 222 -eq 111 ']'              #测试，返回值 222 = 111?
+ echo 'the number 2 is bigger'
the number 2 is bigger
+ exit
```

可以看到，输出显示了每一步代码执行的结果。这对于语法正确、但是输出错误的脚本进行调试时非常有用。

如果一个脚本中存在着语言错误，则用 nv 参数可以方便地检查，下面是一个语法检查的可能结果：

```
/home/earthfans/tmp $ bash -nv bashs.sh
#!/bin/bash
declare -a arr1                   #显式声明一个数组
arr1[1]="abcd"                    #给该数组的第 1 个元素赋初值为 abcd
array[11]=11;                     #直接定义一个数组，并同时将第 11 个元素赋值为 11
for efe in ; do                   #显示两个数组的元素
            $efe
#done
echo "array No.11 is ${array[11]}"
echo "arr1 No.11 is ${arr1[1]}"     #显示未赋值的元素
```

```
echo "unknow arr1 is ${arr[2]}"
echo "unknow array is ${array[2]}"
bashs.sh: line 17: 语法错误：不可预料的文件末尾
```

18.8 小结

本章简单介绍了 Shell 编程的相关知识，通过本章的阅读，虽然不会立刻让读者成为 Shell 编程的专家，但至少可以使读者具有读懂大多数复杂脚本及编写普通脚本的能力。本章介绍的内容涵盖了变量与数组、算术操作符及各种语句的使用、函数等各个方面的内容。在本章的最后，还介绍了脚本调试的简单内容。本章可按顺序进行阅读，在阅读的同时，应该大量的做些脚本编程的练习，Shell 的学习是不断练习的过程。同时通过经常阅读优秀的脚本代码并自己动手编写实现特定需求的脚本，才是提高 Shell 编程水平的最佳途径。

第 19 章　搭建 LAMP 服务器

据统计，在目前的互联网世界中，LAMP 所提供的网络服务占据了 80%以上的市场份额。LAMP（Linux、Apache、MySQL、PHP 或 Perl 或 Python）就是在 Linux 基础上组建的一套功能强大的组合：由 Apache 提供的 WWW 服务器、MySQL 提供的数据库服务、PHP 或 Perl 或 Python 提供的脚本语言支持。这四者的组合，提供的服务不仅性能强劲，而且上述软件均是自由软件，这些大大降低了中小企业和创业者的成本，所以得到了越来越广泛的应用。本章就简单介绍这套组合的安装与简单的设置方法。

19.1　安装 LAMP

Ubuntu 提供了 apt 方式的软件管理工具，使得软件安装与管理方便了许多。同样，新一代的 Ubuntu 也提供了一种套装式安装软件的方法，使得安装一些套件方便了许多。安装 LAMP 就使用了这种套装式安装方法。

19.1.1　安装

为了使用 LAMP 服务，必须先安装相关的服务。在 Ubuntu 中安装是很简单的，只要按照如下步骤进行即可，要注意的是安装前，必须保持与互联网的连接。

（1）单击主菜单“系统”|“系统管理”|“新立得软件管理器”命令，打开新立得软件管理器。

（2）单击菜单“Edit”|“Mark package by task”命令，出现如图 19.1 所示界面。

（3）勾选“LAMP server”项，再单击“OK”按钮即可开始安装。



这种套件方式的安装功能很丰富，参照如上的方法，还可安装其他服务器（如 Tomcat、Samba 等）、2D/3D 软件处理套件、音频处理套件等。选择了安装套件后，系统会处理相关的依赖关系，然后自动安装所有需要的软件。

（4）安装过程中，还会进行基本的设置，如 MySQL 的 root 密码设置。出现图 19.2 所示界面时，只需要在文本框中输入相关密码即可。

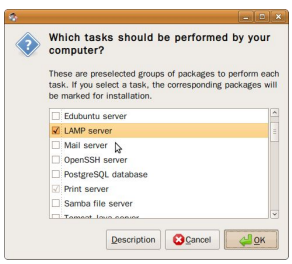


图 19.1　套件安装界面

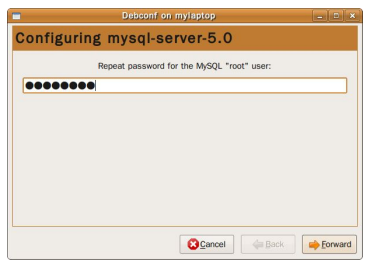


图 19.2　设置 MySQL root 密码

注 意　该密码要牢记，在 MySQL 管理中 root 用户的密码是极其重要的。

（5）软件下载、设置的过程都结束后，“新立得”会报告安装与设置的过程，本例中安装设置的结果如图 19.3 所示。如果在结果中看到有“Sucessfully”或“成功”字样，则说明安装过程中没有发现错误，相关设置也已完成。可以正常运行这些服务了。

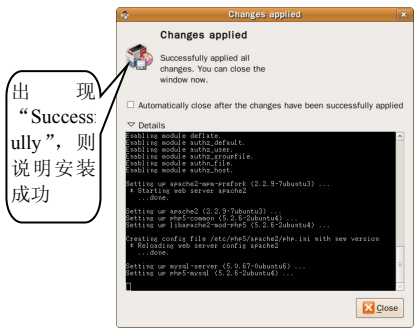


图 19.3　安装结束状

19.1.2　测试

安装完成后，相关的服务（如 Apache、MySQL 等）应该已经启动。要知道这些服务是否已经正常启动，可以通过如下方式检查其状态。

1．查看进程

这些服务运行后，在后台会有相应的进程运行。如 MySQL 的进程名为 mysqld，而 Apache 的进程名为 apache2。所以，可以简单地查看进程名来检查服务是否已经运行。使用命令如下所示：

```
earthfans@mylaptop:~$ ps -e|grep -e "mysql\|apache"
#如果输出的内容类似这样，说明相关的服务已经正常启动
4950 ?        00:00:00 mysqld_safe
4992 ?        00:00:19 mysqld
30691 ?        00:00:00 apache2
30696 ?        00:00:00 apache2
30697 ?        00:00:00 apache2
30698 ?        00:00:00 apache2
30699 ?        00:00:00 apache2
30700 ?        00:00:00 apache2
```

2．测试 Apache 服务器

打开浏览器，在地址栏输入“localhost”。应该可以打开一个网页，上面简单地显示了一行字“It works.”，如图 19.4 所示。这样就说明 Apache 服务运行状态正常。

3．测试 PHP

安装的 LAMP 套件中还包括 PHP。为了检测当前 PHP 系统运转是否正常，以及 Apache 是否已经正常支持 PHP，可以按如下步骤进行检验。

（1）新建一个 PHP 文件，这是个用来测试 PHP 的脚本文件。建立命令如下所示：

```
sudo vi /var/www/info.php
```

（2）在其中加入如下内容，保存文件退出。

```
<?php
phpinfo()
?>
```

（3）打开浏览器，输入“localhost/info.php”，按回车键，打开该网页。输出应该类似如图

19.5 所示。如果出现图 19.5 所示内容，则说明 PHP 功能也运行正常。

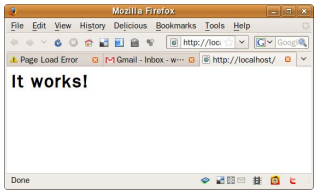


图 19.4　localhost 的默认页面

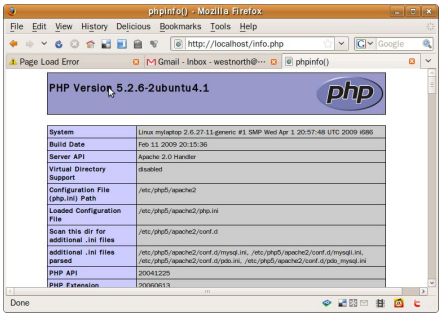


图 19.5　简单的 PHP 页面

4．测试 MySQL

为了检测 MySQL 服务是否运行正常，在终端中输入如下命令：

```
earthfans@mylaptop:~/Documents$ mysql -u root -p
Enter password:
#输入图 19.2 设置的密码
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 29
Server version: 5.0.67-0ubuntu6 (Ubuntu)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>       #如果出现该提示符，说明 MySQL 运行也是正常的。
```

出现“>”提示符，说明已经成功进入 MySQL 环境，可以在其中运行相关的 MySQL 命令来执行相关的功能。

19.2　配置 Apache

LAMP 安装成功后，已经具备了提供简单的 HTML 服务与 PHP 语言处理的能力，但默认的配置并不能完全发挥其能力。为了提高其性能，并且使得 Apache 提供自定义的服务，还需要对其进行一些初步的设置。本节介绍 LAMP 的设置文件所在以及简单的设置方式。

19.2.1　配置文件

LAMP 由 Apache、MySQL 与 PHP 等组成。本节简单介绍一下这些组成部分的主要配置文件。在需要更改配置时，就可在相关配置文件中进行相关的设置。

1．Apache 的默认站点目录/var/www

Apache 默认站点的文件存放目录是/var/www/。也就是说，默认情况下，如果编写了 html 网页或 PHP 动态内容，放在该目录就可以被 Apache 服务器解析，然后在浏览器中打开。如前例所述的 info.php 文件，放在/var/www 目录中，在网络浏览器中以“localhost/info.php”打开，就可以看到 PHP 的函数 phpinfo 的输出内容。

2．Apache 配置文件在/etc/apache2/目录

该目录中，又有一些子目录与文件，用来设置 Apache 的各种特性。将部分子目录及文件的功能列出于下。

- conf.d：该目录存放了两个配置文件 charset，security。charset 用来设置网页的默认字符编码，如果要将默认编码设置为 UTF-8，则只需在其中加入如下内容即可“AddDefaultCharset UTF-8”，而 secutity 是站点的安全相关设置，如网页默认目录的访问权限等相关内容。
- apache2.conf：该文件是 Apache 服务器的全局配置文件，Apache 的一切行为均受该文件的制约。要想在该服务器可以解析域名（如 www.exsample.com），而不是 IP 地址（如 134.25.2.111），就首先需要将大约 169 行的“HostnameLookups Off”更改成“HostnameLookups On”。当然，Apache 的配置不只有这一项，但由于 Apache 的配置是一个极其复杂的过程，所以其余配置在相关内容涉及时再进行介绍。
- ports.conf：该文件配置的是服务器监听端口。默认监听的是 80 端口（http 服务端口）。设置方式是“Listen 80”，如果要监听其他端口，可以照此设置。如“Listen 8080”等。不过，在文件中，每一个端口只能监听一次，也就是说不能有两个“Listen 80”的设置，如果出现重复设置的监听端口。启动 Apache 时会出现无法打开端口的错误提示，导致服务无法启动。
- httpd.conf：Ubuntu 与其他发行版不一样的是，httpd.conf 文件并不是设置站点的配置文件，保留该文件可能也只是为了与其他系统的兼容性。Ubuntu 将站点配置文件按逻辑关系分开放置在许多文件中。如站点相关文件放置在“sites-available”中，安全相关设置在“conf.d/security”中等。
- mods-available 与 mods-enabled：mods-availabl 显示了所有 Apache 可用的模块，包含当前启动的与未启动的所有模块。相应的，mods-enabled 中放置的是当前服务所启用的模块，如目录中有如下文件“php5.load ssl.conf ssl.load php5.conf”，则说明当前服务启动了 PHP 模块与 ssl 模块。
- sites-available 与 sites-enabled：同样，sites-available 包含了所有的站点，而 sites-enabled 则包含了当前启用的站点。理论上来说，可以将所有站点的配置文件均写入 default 文件中，但是这样做会导致站点的管理不便。如果将一个站点写入一个文件，就可以方便地重启或停止单独的站点。

3．Apache 日志在/var/log/apache2/

该目录中的日志包括了访问 Apache 服务器用户的身份等信息，同时还有一些出错信息可供管理员分析。

4．Apache 启动脚本是/etc/init.d/apache2

这是控制 Apache2 的主要脚本，可以取如下参数：

- start：启动服务；
- stop：停止服务；
- reload | force-reload：重新加载数据文件；
- restart：重启服务；
- status：获取服务状态信息；

如“sudo/etc/init.d/apache2 start”是启动服务。

5．PHP 的配置文件在/etc/php5/apache2/php.ini

该文件是 PHP 的主配置文件，可以设置 PHP 的许多基本属性。一些属性的设置方法将在下文予以介绍。

6．MySQL 的数据库存放在/var/lib/mysql

一般来说，MySQL 的数据库文件存放在该目录。备份数据时，可将该目录中的文件复制一份即可。

详细的 MySQL 变量信息可以使用“mysqladmin variables”来查看。使用该命令后的部分输出如下所示：

```
 mysqladmin variables
+---------------------------------+-----------------------------+
| Variable_name                   | Value                       |
+---------------------------------+-----------------------------+
| auto_increment_increment        | 1                           |
| auto_increment_offset           | 1                           |
| automatic_sp_privileges         | ON                          |
| back_log                        | 50                          |
| basedir                         | /usr/                       |
| binlog_cache_size               | 32768                       |
| bulk_insert_buffer_size         | 8388608                     |
| character_set_client            | latin1                      |
| character_set_connection        | latin1                      |
| character_set_database          | latin1                      |
| character_set_filesystem        | binary                      |
| character_set_results           | latin1                      |
| character_set_server            | latin1                      |
| character_set_system            | utf8                        |
| character_sets_dir              | /usr/share/mysql/charsets/  |
| collation_connection            | latin1_swedish_ci           |
| collation_database              | latin1_swedish_ci           |
| collation_server                | latin1_swedish_ci           |
| completion_type                 | 0                           |
```

19.2.2　增加一个基于域名的虚拟主机

Apache 的默认站点目录是/var/www。将自己网站里的文件放置于该目录，就可以在浏览器中输入 localhost 打开这些文件。但这种方式有几个缺点，一是该目录对于普通用户来说，没有写权限。这样会导致一些写操作的失败，从而导致网站运行不正常。二是如果将所有的网站均放于该目录，不便于对于各个网站的分别管理。

本节就介绍一种增加基于域名的虚拟主机的方法，通过该方法可以增加多个主机，每个主机拥有独立的空间与独立的权限。而且更为重要的是，设置的方法很简单，只需要简单的几步即可。在开始之前，应该检查系统，用户系统应具备如下条件。

- 用户所使用的操作系统是 Linux 发行版 Ubuntu，其他操作系统设置与之类似，但有些细微的差别。

- 已经成功地安装了 LAMP 服务器组件，并且这些服务已经运行。
- 本机的 IP 地址是 192.168.1.101。使用其他 IP 地址也可，只需在配置文件中做相应的修改即可。查看本机 IP 地址的命令是 ifconfig。
- 在用户主目录中有一个名为 www 的目录，其中存放着若干 html 或 PHP 文件。目录名为 www 是为了与系统默认目录互相兼容，但该名称并不是必须的，也可以命名为其他名称。里面存放 html 文件与 PHP 文件是为了测试静态与动态网页是否可以正常运行。

而本次设置的目标是建立一个名为 wiki 的虚拟主机，也就是说，建立成功后在浏览器中输入 wiki 就可打开该主机的主页。下面是具体的设置步骤：

（1）用 root 用户身份打开“/etc/hosts”文件，在其中加入“192.168.1.101 wiki”。其中：

- 192.168.1.101：是主机的 IP 地址；
- wiki：是主机名。

整个语句的意思是如果接到对于名为“wiki”的主机的访问请求，就将请求定位至 IP 地址为 192.168.1.101 的计算机（本例中是本机）。

（2）创建配置文件可以自己从头开始，也可以复制一个默认站点的配置文件作为配置网站的基础。我们采用后一种途径，方法如下：

```
sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available/wiki
```



在一个文件基础上进行设置而不是从零开始设置也是一种常用的技巧，这样可以减轻许多设置的基础步骤，也降低出错的概率。

该文件是站点的主配置文件，内容应该如下所示（以“#”开头的语句是注释，为笔者所加）：

```
<VirtualHost *:80>
#使用“*”意思是监听所有 IP 地址的 80 端口。<VirtualHost *.80>的意思是开始一个虚拟主机的配置，在同一个文件中可以有多个虚拟主机的设置。当然我们不推荐在一个文件中设置多个虚拟主机。
    ServerAdmin webmaster@localhost                 #系统管理员邮件
    DocumentRoot /var/www/                          #网站文件的根目录
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>                           #用户目录的相关权限
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow, deny
        allow from all
    </Directory>
    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/         #CGI 脚本位置
    <Directory "/usr/lib/cgi-bin">                  #CGI 脚本权限
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow, deny
```

```
            Allow from all
        </Directory>
        ErrorLog /var/log/apache2/error.log     #错误日志位置
        # Possible values include: debug,  info,  notice,  warn,  error,  crit,
        # alert,  emerg.
        LogLevel warn
        CustomLog /var/log/apache2/access.log combined
       Alias /doc/ "/usr/share/doc/"
       <Directory "/usr/share/doc/">            #网站文档位置
          Options Indexes MultiViews FollowSymLinks
          AllowOverride None
          Order deny, allow
          Deny from all
          Allow from 127.0.0.0/255.0.0.0 ::1/128
       </Directory>
    </VirtualHost>                              #完成一个虚拟主机的设置
```

（3)现在虽然已经建立了配置文件，但如果将其简单地放置在 sites-available 目录中，Apache 服务器是不会读取其中内容的。Apache 服务器只会读取 sites-enable 目录的相关内容。在向 sites-enable 目录中添加文件之前，先来看看该目录有什么内容：

```
ll /etc/apache2/sites-enabled
总用量 0
lrwxrwxrwx 1 root root 26 2008-12-08 15:04 000-default -> ../sites-
available/default
```

可以看到，里面有一个名为 000-default 的文件，而且该文件实际上是 sites-availables 目录中 default 文件的一个连接。这样，我们也就可以参照其做法，将刚才建立的配置文件连接到 sites-enable 目录即可。使用如下命令即可：

```
sudo ln -s /etc/apache2/sites-available/wiki /etc/apache2/sites-enable/wiki
```

（4）下面就可以开始第一个虚拟主机的配置了。修改第二步得到的配置文件，内容如下所示：

```
<VirtualHost *:80>
#同样监听所有 IP 的 80 端口
    ServerAdmin westnorth@gmail.com
#修改系统管理员的邮件地址
    ServerName wiki
#主机名，这里很重要，该名称要和第一步设置的主机名相同。
    DocumentRoot /home/westnorth/www
#文档根目录，也很重要，是放置用户网页内容的所在。
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /home/westnorth/www/>
```

```
    #同样，这儿的目录名也要与 DocumentRoot 设置的目录相同，该部分设置的内容是对于该目录的访问权限
            Options Indexes FollowSymLinks MultiViews
            AllowOverride None
            Order allow.deny
            allow from all
        </Directory>
```

（5）打开刚才设置的网站主目录“/home/westnorth/www”，在里面放置一个普通的 html 文件，如 wiki.html，然后再编写一个简单的名为 wiki.php 文件。两个文件的内容分别是：

```
$cat wiki.php
<?php
phpinfo();
?>
$cat wiki.html
<html>
<body>
<h1>It works!</h1>
this is my world!
</body>
</html>
```

（6）完成设置后，首先使用如下命令强迫 Apache 服务器重新加载设置内容：

```
sudo /etc/init.d/apache2 reload
```

然后，就可在浏览器中分别输入“wiki/wiki.php”与“wiki/wiki.html”，打开这两个文件了。图 19.6 与图 19.7 显示了运行效果。如果看到如图 19.6 与图 19.7 所示的效果，说明名为 wiki 的虚拟主机已经安装成功。当然，还可以参照以上步骤，建立更多的虚拟主机。以适应不同的用途。

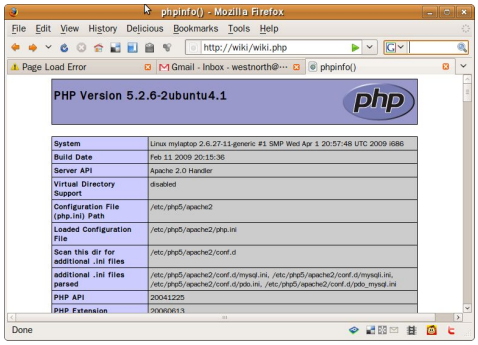


图 19.6　wiki/wiki.php 的显示效果



图 19.7　wiki/wiki.html 的显示效果

19.2.3　使用 SSL 加密网页传输

网页通信中，有些网页内容是比较敏感的，如涉及商业机密或个人用户的银行账户与密码等内容。这时，采用普通模式的网页传输就显得不安全了。这时，就可使用 SSL（Secure Sockets Layer）“安全套接层”来保证网络传输的安全。

1．启用 SSL 模块

在 Apache 中使用 SSL 协议也是件很容易的工作。启用 Apache 的 SSL 模块只需以下几步即可。

（1）由于在前面已经安装了 Apache 的 SSL 模块，这里只需要启用该模块即可。启用 ssl 模块的命令如下所示：

```
sudo a2enmod ssl
```

其中 a2enmod 是专用于启动 Apache 相应模块的程序，还有一个与之功能相反的屏蔽模块的程序 a2dismod。

（2）可以使用 openssl 程序来创建 SSL 证书，创建证书的方法是：

```
earthfans@mylaptop:/etc/apache2$ sudo openssl req -x509 -newkey rsa:1024
-keyout apache.pem -out apache.pem -nodes -days 999
#在使用该命令后，会出现交互模式，提示用户输入相关内容，出现的内容应该如下所示：
Generating a 1024 bit RSA private key
..................++++++
.................................++++++
writing new private key to 'apache.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.'. the field will be left blank.
-----
Country Name (2 letter code) [AU]:CN
#国家名称，这里输入的内容应该类似于网址中国家的代码，如中国则应该输入 CN
State or Province Name (full name) [Some-State]:Shanghai        #输入省名
Locality Name (eg, city) []:Shanghai                            #城市名
Organization Name (eg, company) [Internet Widgits Pty Ltd]:     #公司名
Organizational Unit Name (eg, section) []:                      #组织名
Common Name (eg, YOUR name) []:westnorth
#这里是整个输入中最重要的内容，输入内容应该是运行 Apache 服务器的主机名，可以使用
hostname 命令来得到本机的主机名。
Email Address []:westnorth@gmail.com
#管理员的邮件地址，该内容会出现在相关的配置文件中。同时该内容也是配置的最后一步。输入
邮件地址再回车后，就完成了创建过程。
```

查看当前目录，会发现其中多了一个名为 apache.pem 的文件，这就是我们需要的 SSL 证书。

2．配置网站

得到了 SSL 证书文件后，就可以在网站配置中使用该证书来签署网页了。下面是一个简单的使用 SSL 的站点配置过程：

（1）复制一个 Apache2 站点的默认文件，我们再在其基础上加入 SSL 支持即可。使用如下命令：

```
sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available/ssl
```

（2）将该文件连接至 sites-enabled 目录，使得 Apache 可以加载该站点。使用如下命令：

```
ln -s /etc/apache2/sites-available/ssl /etc/apache2/sites-enabled/ssl
```

（3）在站点中加入 SSL，主要是配置刚才复制的站点文件即可。使用如下命令打开配置文件：

```
vi /etc/apache2/sites-available/ssl
```

这条命令用于打开配置文件。只需在该文件中将网站的监听端口更改为 443，然后加入相关的 SSL 认证文件即可，其他部分与普通网站配置是一致的。

（4）加入了 SSL 认证及更改了端口的文件内容应该如下所示：

```
#文件：/etc/apache2/sites-available/ssl
<VirtualHost *:443>
        ServerSignature On
        SSLEngine On
        SSLCertificateFile /etc/apache2/ssl/apache.pem
#这里的路径是实际文件的保存路径，可以修改为保存有 apache.pem 的路径，也可以将该 SSL 证书复制到该目录中，如果这里设置的目录中找不到 apache.pem 文件，Apache 启动时也会报错。
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/
        <Directory />
               Options FollowSymLinks
               AllowOverride None
        </Directory>
        <Directory /var/www/>
               Options Indexes FollowSymLinks MultiViews
               AllowOverride None
               Order allow，deny
               allow from all
        </Directory>
        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
               AllowOverride None
               Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
```

（5）再检查一下 ports.conf 文件，该文件配置了 Apache 的端口，确保其中设置了监听 443 端口（SSL 的端口）。用任何一款文本编辑器（如 vi）打开该文件，要确保该文件中有且只有一行内容是 Listen 443：

```
$vim /etc/apache2/ports.conf:
//下面的内容是 ports.conf 文件的内容
NameVirtualHost *:80
Listen 80
#Listen 443
#如果其他地方没有 Listen 443，则将此处的#删除。否则就保留该注释符号，如果配置文件中有
```

两个监听 443 端口的设置，则 apache 系统启动时会报错，而无法启动。

```
<IfModule mod_ssl.c>
    # SSL name based virtual hosts are not yet supported, therefore no
    # NameVirtualHost statement here
    Listen 443
</IfModule>
```

3．重启服务器

配置完成后，应该强制 Apache 重新载入或重启 Apache 服务来使设置生效。重新载入及重启 Apache 的方法如下。

- 重新载入 Apache 配置的命令是“sudo /etc/init.d/apache2 force-reload”。
- 重启 Apache 服务的命令是“sudo /etc/init.d/apache2 restart”。

19.3　MySQL 基本操作

MySQL 是一款自由的数据库软件，其提供的服务速度快，性能可靠，已经成为许多大型网站的首选。经过实际的生产应用，Windows 的 NTFS 文件系统与 Linux 系统中的 ext3 等文件系统均可应用在高负荷的商业环境中。本节就介绍一些数据库的简单操作。

19.3.1　连接数据库服务器

要连接服务器，首先需要得到服务的授权，也就是说，用户需要一个可能使用的用户名与密码来连接。连接的方法是：

```
mysql -h host -u user -p
```

- -h 后面跟数据库服务器所在的主机名或 IP 地址。这就是说，MySQL 可以进行远程登录。如果连接的是本机的数据库服务器，则可以不使用该参数。
- -u 后面跟数据库用户名
- -p 意思是该用户需要输入密码。

下面以一个实际的使用示例来演示 MySQL 的用法：

```
mysql -u root -p
Enter password:
#服务器要求输入密码，输入正确的密码后就可以连接到数据库服务器。
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 5.0.67-0ubuntu6 (Ubuntu)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
#一些提示信息，如果出现“mysql>”提示符，说明连接成功。
```

19.3.2　MySQL 命令简介

MySQL 提供了种类繁多的命令，可以完成数据库管理的所有任务。这些命令均是在 MySQL 命令提示符后输入的。由于 MySQL 命令种类极其繁多，所以这里只简单介绍一下命令的格式。

下面是一个把 MySQL 当做计算器的示例：

```
mysql> select sin(pi()/4), (4+1)*5;
#在没有连接到数据库前，先介绍一种使用技巧，可以将 mysql 当做计算器来用，如输入上面的表达式，输出会如下所示：
+------------------+---------+
| sin(pi()/4)      | (4+1)*5 |
+------------------+---------+
| 0.70710678118655 |     25 |
+------------------+---------+
1 row in set (0.00 sec)
```

上面也演示了 MySQL 命令的格式。一般情况，MySQL 命令由一条 SQL 语句组成，最后是一个“;”，代表命令的结束。如：

```
mysql> select version();
+-----------------+
| version()       |
+-----------------+
| 5.0.67-0ubuntu6 |
+-----------------+
1 row in set (0.00 sec)
```

MySQL 可以接受命令的分行输入，输入命令的一部分回车，并不提示输入错误，而是一直等待用户的持续输入，输入“;”MySQL 才将输入的命令执行。在输入命令的过程中，可以输入“\c”来取消当前输入。

数据库的操作是一件非常复杂的工作，涉及许多方面。而且由于数据库在许多应用程序中的重要性，操作数据库必须谨慎小心。下面介绍了一些操作数据库的简单操作。如果需要完整的命令列表，应该查阅 MySQL 手册。下面的命令均运行在 MySQL 命令提示符中。

1．显示当前数据库

一般来说，每个数据库程序均存在多个数据库。可以使用不同的数据库来完成不同的功能。查看当前使用的数据库的命令如下所示：

```
mysql> show databases;          #显示当前使用的数据库
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
+--------------------+
2 rows in set (0.00 sec)
```

输出显示当前使用了两个数据库，分别是 information_schema 与 mysql。

2．创建数据库

但有时也会出现当前的数据库无法满足使用要求的情况。这时就可新建一个数据库来完成相关的功能。新建数据库的命令是：

```
mysql> create database ubuntu;
Query OK， 1 row affected (2.57 sec)
```

可以使用 create database 命令来创建数据库。要注意的是虽然 MySQL 命令是不区分大小写的，但数据库名称是区分大小写的。如 myDatabase 与 MyDataBase 是两个不同的数据库。

3．使用数据库

要从当前的数据库切换到另一个数据库，可以使用命令 use。示例如下：

```
mysql> use ubuntu;                    #切换到数据库 ubuntu
Database changed                      #回显信息说明已经将当前数据库更改
```

使用 use 命令可以更改当前的数据库。也可以在终端中连接时指定要使用的数据库。示例如下：

```
mysql  -u user -p ubuntu              #使用用户 user 操作数据库 ubuntu
```

要注意的是，-p 参数后跟的并不是该用户的密码，如果要在命令中使用密码，则密码应该紧跟在-p 后，不能有空格，如“-pwe3e”。

4．创建表

假设现在已经进入了新创建的数据库 ubuntu，但新创建的数据库是没有任何内容的。可以通过 create table 命令来新建表。示例如下所示：

```
mysql> CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20), species
VARCHAR(20), sex CHAR(1), birth DATE, death DATE);
#建立一个名为 pet 的新表
Query OK, 0 rows affected (2.51 sec)
```

5．向表中添加数据

新表建立后，就可向其中添加数据了。添加数据的命令如下所示：

```
mysql>  INSERT INTO pet VALUES ('Puffball','Diane','hamster','f','1999-03-
30',NULL);
#使用 INSERT INTO 命令向 pet 表插入一行数据
Query OK, 1 row affected (0.01 sec)
```

6．显示数据

要查看表中已经存在的记录，可以使用 SELECT 命令进行查看。示例如下：

```
mysql> SELECT * FROM pet WHERE name = 'Puffball';
#查询表 pet 中 name 为 Puffball 的宠物信息
+----------+-------+---------+------+------------+-------+
| name     | owner | species | sex  | birth      | death |
+----------+-------+---------+------+------------+-------+
| Puffball | Diane | hamster | f    | 1999-03-30 | NULL  |
+----------+-------+---------+------+------------+-------+
1 row in set （0.00 sec）
```

7．创建用户

可以使用 grant 命令来创建用户。如下面的命令是创建一个名为“westnorth”的用户，其登录密码是“123456”，该用户对数据库 Ubuntu 有所有的权限：

```
mysql>GRANT ALL PRIVILEGES ON ubuntu.* TO 'westnorth'@'localhost'
IDENTIFIED BY '123456';
```

8．删除用户

可以使用 DROP USER 命令来删除用户，该命令同时也会取消对于该用户的所有授权。如果该用户已经登录的话，删除用户的操作会等到用户退出登录后才会生效。如：

```
DROP USER westnorth
```

该命令删除刚创建的用户 westnorth。

9．备份数据库

官方建议备份数据库时使用 mysqldump 来进行，如下所示：

```
mysqldump ubuntu > /home/westnorth/sample.sql
```

功能是将数据库 ubuntu 备份至 sample.sql 文件。如果数据库文件比较大，也可以在备份前对其进行压缩：

```
mysqldump ubuntu | gzip > /home/westnorth/sample.gz
```

10．恢复数据库

恢复数据库很简单，只需要将备份的文件作为 mysql 的输入即可。如：

```
mysql ubuntu < sample.sql
```

就可将 sample.sql 文件恢复至数据库 ubuntu 中。



MySQL 的命令与标准 SQL 语言是兼容的，如果需要更多的 MySQL 操作命令，可以查阅诸如“标准 SQL 语言教程”之类的教材获得完整的命令列表。

19.4 PHP 基本设置

PHP，是一个嵌套的缩写名称，英文全称是（PHP:Hypertext Preprocessor）。PHP 是运行在服务器的一种脚本语言，其速度比 CGI 或者 Perl 更为快速。PHP 具有非常强大的功能，可以实现其他所有脚本语言的功能，如 CGI 或者 JavaScript，而且支持几乎所有流行的数据库以及操作系统。

PHP 的配置文件是/etc/php5/apache2/php.ini 文件，在修改配置文件之前，应该将文件先行备份。一般而言，配置文件的内容中可以修改的内容如下。

1．memory_lim it=20M

内存限制，越大则 PHP 的性能越好，但可能会影响其他程序的内存使用，可以参照自己的情况修改为所需的内存大小。

2．upload_max_filesize=10M

最大上传文件尺寸，这是使用 PHP 的程序的通用限制。凡是使用 PHP 语言的站点，在 PHP 页面中可上传的文件大小由该值决定。

3．register_long_arrays=On

应该打开该选项。因为目前大多数 php 的代码都是用 php4 写的，为了兼容以前的 php 版本，需要打开本选项。

4．safe_mode=On

打开安全模式，打开该选项可以增强 PHP 系统的安全性，堵住类似 php shell 之类的后门。不过，打开该选项会导致 judege online 一类调用外部程序的 PHP 代码无法正常运行。

5．open_basedir=/var/www/

限制了 fopen（），file（）等函数只能打开“/var/www/”目录，该选项只允许运行/var/www/目录下的 PHP 文件。

6．disable_functions=phpinfo，get_cfg_var

禁止使用 phpinfo 和 get_cfg_var 函数，这样可以保护本机的服务信息不被 phpinfo 之类的函数泄露出去。可以在 19.3 节介绍的 php 网页正常运行后，设置该项，然后重启服务，再运行该 PHP 网页，就可以看到 phpinfo（）函数已经无法获得相应的信息了。

7．打开错误日志

错误日志功能默认是关闭的，可以通过修改如下语句将其打开。这样，就可以查看 PHP 的出错信息，进行相关的处理了。

```
error_reporting = E_ALL
log_errors = On
error_log = /var/log/php_err.log
```

8．register_globals=Off

;如果清楚 php 程序不需要 register_globals，就将其关掉。如果清楚 php 程序需要，则将其打开。Ubuntu 默认是关闭的。

9．allow_url_fopen=Off

禁止打开远程地址。可以避免一些安全方面的问题。

19.5　安装 Web 应用程序

现在，一个性能强大的 LAMP 系统已经搭建成功，但是如果没有功能丰富的应用程序的话，这些强大的功能是得不到发挥的。本节将简单介绍两个在互联网世界中得到广泛应用的 Web 应用程序的安装过程，以使读者对在 LAMP 系统安装 Web 应用程序有一个初步的了解，然后就可安装更多的应用程序来丰富其功能。

19.5.1　安装 MediaWiki 维基百科

在互联网世界里，有一个内容丰富的自由百科全书站点，该站点收集了人类几十种语言的几千万份文档，其资料规模远远超过著名的传统权威大英百科全书。其内容包罗了几乎所有的知识门类。而且其有个最大的好处是，用户可以自由编辑其内容。也就是说，如果你对某一方面了解颇多，发现该网站提供的信息不准确时，可以随时编辑其网页内容。而所有这一切，均是免费提供的。这个站点的名字就称为维基百科（网址是 http://www.wikipedia.org）。

现在维基百科在世界范围内已经成为一个权威的参考书，许多人遇到不知道的事物，往往第一反应是去查维基百科。虽然可能由于语言的差异，再加上种种原因，维基百科在中国的流行程度并不是很广，但其影响还是越来越大。而这个维基百科，构建其的基础就是 MediaWiki 软件。如果系统中安装了 LAMP 系统，那么安装 MediaWiki 也是一件很容易的事，只需简单的几步即可。下面介绍安装的相关知识。

1．获得源代码

使用 wget 命令获得 MediaWiki 源代码，当然你也可以使用其他任何常用的下载软件下载该源码包。使用命令如下：

```
wget http://nchc.dl.sourceforge.net/sourceforge/wikipedia/mediawiki-1.5.4.tar.gz
```

解压该包，使用如下命令：

```
tar xvzf mediawiki-1.5.4.tar.gz
```

得到一个名为 mediawiki-1.5.4 的目录，该目录包含了所有 wikipedia 的文件。

2．前期设置

在安装 MediaWiki 系统之前，还需要对系统进行相关的设置。如设置一个专用于处理 wiki 的虚拟主机，以及在 MySQL 中创建用于 MediaWiki 操作的数据库与用户。最好再安装一款用于处理图片的软件，可以创建缩略图，以及给图片添加水印等功能。

（1）为了使 MediaWiki 能够作出一些图片的特效，应该在安装 MediaWiki 之前安装一款图形软件：imagemagick，方法是：

```
sudo apt-get install imagemagick
```

（2）将源码包中 config 目录的权限设置为可写，实际上对于 mediawiki-1.5.4 目录，也应该设置可写权限。

```
chmod a+w mediawiki-1.5.4/config
```

（3）使用 root 用户登录进 MySQL，然后进行如下的操作：

```
mysql> create database wiki;
#创建用于放置 wiki 内容的数据库 wiki，这个名称是任意的。只要在 MediaWiki 的设置中使用该名称即可。
mysql> grant all privileges on wiki.* to 'westnorth'@'localhost' identified by '123456';
#创建一个操作该数据库的用户“westnorth”，并且设置其密码是 123456.
mysql> flush privileges; 刷新用户授权表。
```

（4）参照 19.2.2 节内容设置虚拟主机，如主机配置文件的部分内容可能类似这样：

```
ServerName wiki
DocumentRoot /home/westnorth/www/mediawiki-1.5
```

3．wiki 的安装

在前面的准备工作就绪后，就可以开始 wiki 的安装过程了。具体步骤如下所示。

（1）在浏览器中输入 wiki（这个要按照虚拟主机的相关设置而改变），就可直接打开 wiki

的主设置页面了。该页面应该如图 19.8 所示。



图 19.8　wiki 初始设置页面

（2）单击其中的“set the wiki up”链接，进入 Mediawik 设置页面。该页面是 MediaWiki 的主设置页面，里面的选项比较多，将其内容列出如下。

- Site name：wiki 站点的名称。不能是空或者取 "MediaWiki"，而且名称中最好不要有标点符号。
- Contact e-mail：管理员的电子邮件地址。
- Language：站点语言，推荐选择英文，这样虽然可能不太直观，但为后来的设置带来了便利。
- Copyright/license metadata：版权类型，站点内文章的授权方式。
- Sysop account name：系统管理员账号。默认是 WikiSysop。
- Password：系统管理员密码。
- Again：再输入一次管理员密码。
- Shared memory caching：内存缓存模式。可以给站点加速。
- MySQL server：如果 MySQL 运行在本机上，就可使用默认值 localhost，否则，就要输入相应的 MySQL 服务器的地址。
- Database name：数据库名称，要与第 2 步建立的数据库名称相同。
- DB username：数据库的用户名，参照第 2 步的设置。
- DB password：数据库的用户密码。
- Again ：再输入一次数据库的用户密码。
- Database table prefix：数据库中表的名称前缀。如果要更好地与其他系统兼容，可以不使用前缀。
- Database charset：数据库编码。
- DB root password：数据库根用户的密码。如果前面输入的数据库在服务器中不存在的话，就要提供数据库的根用户密码，以便创建该数据库。如果数据库服务器中有该数据库的话，就可以不用输入密码。

（3）设置完成后，就可单击“Install”按钮开始安装了。浏览器会出现如下内容：

```
    # Trying regular user... ok.
    # Connected to 5.0.67-0ubuntu6; enabling MySQL 4 enhancements
    # Database wiki exists
    # Creating tables... using MySQL 3/4 table defs... done.
    # Initializing data...
    # Created sysop account horseman.
    #
    Initialising "MediaWiki" namespace...
    Clearing message cache...Done.
    Creating LocalSettings.php...
    Installation successful! Move the config/LocalSettings.php file into the
parent directory, then follow this link to your wiki.
```

如果最后出现“Successful”字样，说明安装成功。

（4）在终端中进入 wiki 的主目录，将 config 目录中的 LocalSettings.php 文件上移一层到 wiki 的主目录，可能像这样：

```
cp mediawiki-1.5.4/config/LocalSettings.php mediawiki-1.5.4/
```

（5）单击第 3 步显示的浏览器页面最后一行的“this link”超链接就可进入 wiki 的主页面了。

（6）在浏览器中刷新 wiki 的主页，就可以看到维基百科的主页面。

（7）在 searching 文本框中输入要添加到 wiki 的内容，单击“Go”按钮就可开始创建相关的 wiki 页面了。

维基百科（www.wikipedia.org）提供了一种方式，可以将其中的文章内容数据库打包下载，然后在本机组建一个本地维基百科。这样对于我国用户来说使用起来就方便多了。而且，其文件也不太大，中文页面由于内容较少，只有大约 200MB，而英文百科也只有 1GB 左右。具体的导入方法，可以在维基百科的站点上进行查询。导入后的维基百科主页界面如图 19.9 所示。

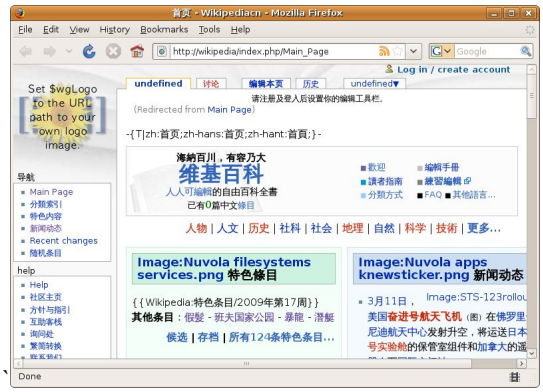


图 19.9　wiki 站点主界面



维基百科是一个多人协作程序，可以用于公司内部网的公司知识库的建立，或者相关技术论坛的知识库的建立，还可用于个人的笔记记录工作等。该程序已在各种行业得到越来越多的应用。

19.5.2　安装 WordPress 博客程序

现在，写博客已经成为一种潮流，名人可以用博客来赚取更高的人气，而更多人的其作为一种交友、书写人生感悟及记录日常琐事的手段。虽然各大网站均提供了博客的空间，但大多数空间对于各种使用限制过多，所以越来越多的人开始使用独立博客，可以不受制于任何限制（除了中国法律），给了用户极大的自由选择权。

而所有的博客程序中，最为著名、使用最广泛的可能就要算是 WordPress 了。其是一款基于 LAMP 系统的博客程序，性能优良，而且现在已经由开发者开发出了种类繁多、功能丰富的插件，更使得其如虎添翼。就像 Linux 世界的大多数软件一样，其是开放源码的自由软件。最后但并非不重要的一点是，WordPress（有时也被简称为 WP）号称是最容易安装的博客系统。安装步骤如下。

（1）获得源码，使用如下命令：

```
wget http://wordpress.org/latest.tar.gz
```

（2）解压源码到/www 目录，使用如下命令：

```
tar -xzvf latest.tar.gz -C ～/www/
```

将其解压到用户主目录的 www 子目录，会产生一个～/www/wordpress 目录。

（3）参照前面所述内容设置虚拟主机，在虚拟主机设置中，将站点目录定位到～/www/wordpress。设置站点名称为 blog。

（4）使用 root 用户登录到 mysql 数据库，然后进行如下操作：

```
mysql> CREATE DATABASE wordpress;
#创建用于 WordPress 的数据库
mysql> grant all privileges on wordpress.* to 'westnorth'@'localhost'
identified by '123456';
#设置用户 westnorth 对数据库的权限。
mysql> FLUSH PRIVILEGES;
#刷新用户权限表。
```

（5）进入～/www/wordpress 目录。使用如下命令：

```
cd ～/www/wordpress
```

（6）将名为 wp-config-sample.php 文件重命名为 wp-config.php，使用命令：

```
mv wp-config-sample.php wp-config.php
```

（7）打开 wp-config.php 文件，进行如下设置：

```
define('DB_NAME', 'wordpress');         // The name of the database
//设置数据库名称，注意本文件中所有设置项应与第 4 步内容相对应
define('DB_USER', 'westnorth');         // Your MySQL username
//设置用户名
define('DB_PASSWORD', '123456');        // ...and passwo
//设置用户密码
```

（8）打开浏览器，输入“blog”，就可打开如图 19.10 所示的 WordPress 安装界面。



图 19.10　WordPress 安装界面

该安装界面显示了安装 WordPress 所需的一些信息，按照提示输入相应的内容即可。如果对这些选项的意义不太理解，可以参见下文的介绍：

“是否允许搜索引擎搜索站点”的意思是如果该博客需要更多的人浏览，就该允许搜索引擎搜索该站点。而如果该站点有一些不足以为外人知道的内容，想要对外界保密，就应该取消该选项。

（9）按图示进行设置后，单击“Install WordPress”按钮开始系统设置工作。设置速度很快，一般也就是几秒种的时间，设置结束后，会出现图 19.11 所示的用户名与密码。

（10）用户 admin 是 WordPress 系统管理员，而密码是采用一定算法随机生成的，应该将其记下来，以备使用。记录下密码后单击“Log In”按钮登录 Wordpress。登录界面如图 19.12 所示。

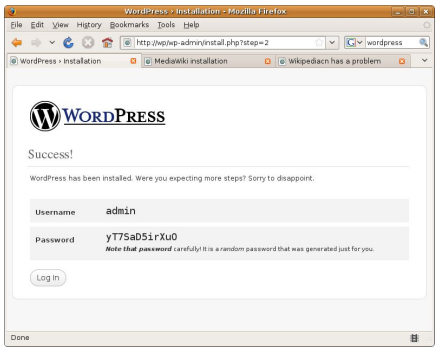


图 19.11　WordPress 设置成功后生成的随机密码

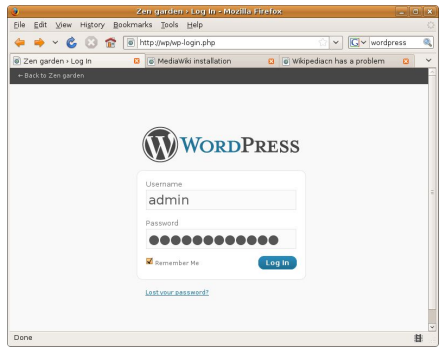


图 19.12　登录界面

（11）输入用户名一密码登录成功后，就可在博客中发表文章了。WordPress 默认的界面很漂亮，发表了一篇博客的 WordPress 界面应该如图 19.13 所示。



图 19.13　博客主页面



如果购买了独立的空间与域名，而且有精力维护自己的博客。可以将自己建立的博客文件上传到购买的空间，设置好相关的数据库后，就可以让网络上的其他用户阅读自己的博客。

19.6　小结

本章介绍了搭建 LAMP 服务器的相关知识，并以“维基百科”与“博客程序”为例介绍了安装 Web 应用程序的方法。要搭建一个安全、高性能的 LAMP 服务器，相关设置是很重要的一项内容。所以本章关于 Apache、MySQL、PHP 等项目设置的相关内容是很重要的，只有熟练掌握了这些设置内容，才能建立高效的服务器。本章的难点内容是 19.2.2 小节“增加基于域名的虚拟主机”，该小节设置项目繁多，概念不容易理解，应该多加练习予以掌握。

第 20 章　FTP 服务器的搭建

FTP 是一种古老但是却仍然焕发着青春活力的技术。一般上网浏览网页资源其实服务器端使用的是 HTTP 协议，而大多数网站提供的文件、资料普通采用的是 FTP 协议。也就是说，要寻找资源的话，在 FTP 站点得到的资源要远多于在 HTTP 站点中所能得到的。而且在局域网中，也普遍采用 FTP 作为内部传输的手段。所以了解和掌握 FTP 的基本知识是很有必要的。

20.1　FTP 服务器安装

FTP 服务器就是在网络中提供 FTP 服务的计算机。而组建 FTP 的过程其实就是向服务器安装 FTP 服务器软件的过程。现在可供选择的 FTP 服务器软件有许多，本节以 Ubuntu 默认的 FTP 服务器软件 WU-FTP 为例介绍了其安装过程。当然，为了更好地使用 FTP，在本节还介绍了 FTP 的相关知识。

20.1.1　FTP 简介

FTP（File Transfer Protocol，文件传输协议）于 1985 年由 J.Postel、J.Reynold 提出，在 http://www.ietf.org/rfc/rfc959.txt 有该协议的文本。FTP 在互联网世界中是一个很重要的基础协议，网络上有众多的 FTP 服务器，提供了数量众多的软件、资源以供下载。

1．使用 FTP 服务

FTP 服务的一般使用方法是使用 FTP 软件连接到提供 FTP 服务的远程计算机。使用 FTP 软件连接的计算机称为客户机，而提供 FTP 服务的远程计算机称为服务器。服务器可以分为系统服务器与匿名服务器，系统服务器只允许有系统账号的用户进行登录，而匿名服务器允许没有系统账号的用户以匿名方式登录。客户登录到服务器后，就可下载服务器允许下载的内容，如果拥有权限，还可将文件上传至服务器。

如果拥有个人网站，应该会使用 FTP 软件来管理远程网站的经验。一般管理远程网站的内容均使用 FTP 方式，因为 FTP 可以自动检查，只更新发生改变的文件，可以实现批量上传、下载等这些对于网站管理来说是极为重要的。

2．登录到 FTP 站点

现在大多数的网络浏览器与文件管理器均具有 FTP 连接的能力。只要获得一个 FTP 服务器的地址，就可以尝试连接到该服务器。一般的 FTP 服务器地址类似于 FTP://23.182.110.211 或 FTP://FTP.ubuntu.org.cn。如在浏览器中输入 FTP 服务器地址，则会打开该 FTP 站点，如果 FTP 站点需要密码的话，还会提示输入密码。用户也可以在地址栏中输入此格式的地址 FTP://用户名:密码@FTP 站点地址，如“FTP://tom:123sew@FTP.ubuntu.org.cn”，就可直接打开要求密码的 FTP 站点。

3．主动模式与被动模式

FTP 数据传输有两种模式：主动模式与被动模式。主动与被动是对于服务器而言的。主动模式中，要求客户端打开一个端口，然后开始监听端口，并且告诉服务器自己打开的端口，然后服务器打开 20 端口并且主动建立与客户机的连接，所以被称为主动模式。在被动模式下，服务器只是打开相应的端口，并进行监听，如果有客户要求连接，服务器接受连接，如果客户再要求传输文件，则服务器开始传输文件。对于服务器而言，连接与传输均是被动的，所以称为被动模式。

主动模式由于要求客户端打开端口，所以对于在局域网内或者打开防火墙的情况下可能无法使用，被动模式在防火墙打开的情况下也能很好的工作，但是有些 FTP 服务器并不支持被动模式。

4．文件类型

连接到服务器后，最主要的工作就是文件传输。但在实际的传输前，还有一点预备工作，就是要设定传输的文件类型，这是因为不同的文件类型，对于 FTP 来说传输方式是不一样的。FTP 将文件分为四种类型。

- ASCII 文件类型：发送时，要传输的文本文件被转换为 ASCII 格式，每行以 CR/LF 结束，接收端接收文件后，再将这些 CF/LF 删除，转换为本地文件。
- EBCDIC：与 ASCII 类似，但是由 EBCDIC 格式进行编码。
- 二进制文件：完全保持文件内容格式进行传输。
- 本地文件系统：用在一些特殊环境。

5．文件传输模式

文件的传输也有如下三种不同的模式。

- 流模式：将文件以字节流的方式传输。
- 块模式：将文件分成特定结构的块，然后以块为单位进行传输。
- 压缩模式：简单地压缩传输的内容。

20.1.2　FTP 服务器软件安装

Linux 世界有许多 FTP 服务器软件，如 WU-FTP（Washington University FTP），VSFTP（Very Secure FTP）等。WU-FTP 作为 Linux 世界中一款历史悠久的软件，市场占有率是最大的，如 Ubuntu Server 版 FTP 服务器套件默认安装的就是 WU-FTP。本节以 WU-FTP 为例介绍其安装步骤。具体步骤如下。

（1）Ubuntu 的软件源中有 WU-FTP，所以可以直接通过 apt 来安装。在终端使用“sudo apt-get install wu-ftpd”就可安装，如果喜欢使用图形界面进行安装，可以打开“新立得软件管理器”搜索“wu-ftpd”，然后选择安装即可。

（2）如果当前运行在级别 2（Ubuntu 默认的运行级别），则会在/etc/rc2.d 目录中生成一个 S050wu-ftpd 文件。该文件链接自/etc/init.d/wu-ftpd，用于在系统启动时运行 WU-FTP 服务。

（3）使用命令“ps -e|grep wu-ftp”检查一下，如果出现类似如下所示的结果，说明服务已经运行：

```
5123 ? 00:00:00 wu-ftpd
```

（4）如果第 3 步检查出 FTP 服务没有运行，则可使用如下方法手动启动该服务：

```
sudo /etc/init.d/wu-ftpd start
```

（5）启动服务后，就可在其他计算机上进行登录了。假设当前计算机 IP 地址为 192.168.1.113，有一个名为 user 的用户，其密码为 123456。这就可在另一台计算机上以如下命令进行登录：

```
lftp user:123456@192.168.1.113
```

20.2 图形界面软件 FileZilla

FileZilla 是 Linux 中一款功能强大的 FTP 软件。其不只可以运行在 Linux 中，其还拥有 Windows、Mac OS 等各种平台的版本。而且其不只是一款 FTP 客户端，同样也可作为 FTP 服务器软件来使用。本节主要介绍其作为客户端时的用法。

1. FileZilla 的主界面

FileZilla 的主界面如图 20.1 所示。各部分的功能分别介绍如下。

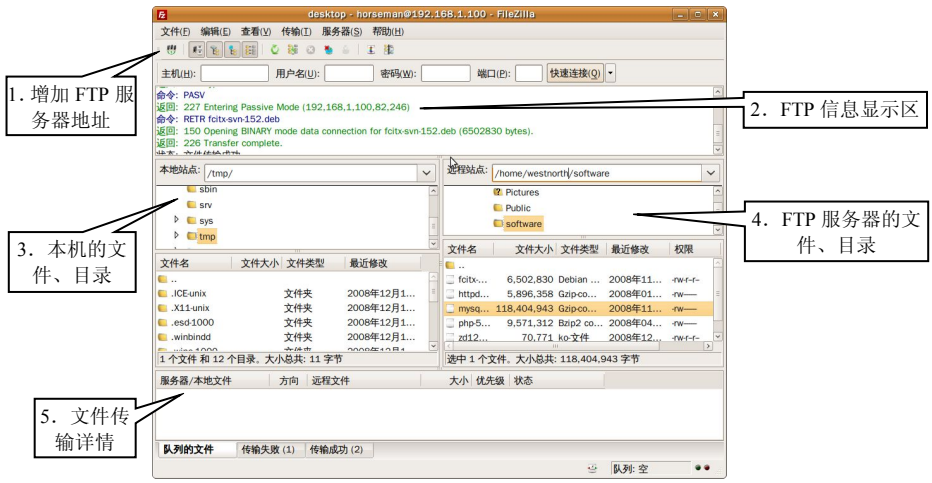


图 20.1 FileZilla 界面

- 增加 FTP 服务器地址：FileZilla 可以管理多个 FTP 站点。要将多个 FTP 站点加入，就可以按下 Ctrl+S 组合键或者单击“增加 FTP 服务器地址”按钮，增加 FTP 站点。还可在其中填写用户名与密码。下次要连接到 FTP 时，就可在站点管理器中单击站点完成自动连接，不用输入用户与密码。
- FTP 信息显示：显示了与服务器连接时的状态信息或文件传输的详细信息。如果出现问题，可以从该区内容判断问题产生的原因。
- 本机的文件、目录：FileZilla 将主界面中间部分分为两部分。左边的显示了本机的文件与目录，这样就可以方便地定位到要上传到 FTP 服务器的文件。
- FTP 服务器的文件、目录：显示了 FTP 服务器中可供用户使用的文件目录，用户可以定位文件上传的位置或者寻找到下载的文件。
- 文件传输详情：显示了文件传输（上传到 FTP 服务器或由服务器下载的文件进度列表）。

2．设置 FTP

在开始连接到第一个 FTP 站点前，还应该对 FileZilla 进行初步的设置，以更好地适应用户的需求。

单击菜单“编辑”|“网络设置向导”，系统会提供一个向导来完成设置过程，用户只需要根据提示选择相关选项即可完成该向导。向导的其中一项如图 20.2 所示，其余页面也与之类似，只需要选择相关的选项，然后单击“Next”按钮即可完成设置。设置完成后，FileZilla 会根据用户的选择给出一个测试结果。

单击菜单“编辑”|“设置”命令，打开如图 20.3 所示对话框，可以在其中进行 FTP 的一些基本设置。

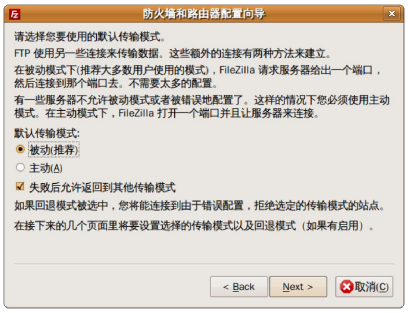


图 20.2　网络设置向导



图 20.3　FTP 基本设置

可以在其中设置传输模式，如主动、被动模式、还可以进行 SFTP（安全 FTP）的连接、在文件类型中可以选择文件的传输类型。可以在“把以下的文件类型作为文本格式文件进行传输”列表框中加入要作为文本文件的文件类型。还有许多其他设置，如数字格式显示、日期格式等，可以打开选项进行查看，这里就不再赘述了。

3．连接到 FTP 站点

单击如图 20.1 所示“增加 FTP 服务器地址”按钮或者按下 Ctrl+S 组合键，打开如图 20.4 所示站点管理对话框。然后参照如下步骤，即可完成 FTP 站点的新建工作。



图 20.4　站点管理

（1）单击“新站点”按钮，新建一个 FTP 站点。

（2）在“主机”文本框中，输入 FTP 站点的 IP 地址或者域名。

（3）在端口中输入 21（FTP 默认端口）。

（4）服务器类别选择“FTP”。

（5）登录类别选择“账号”。如果 FTP 服务器可以进行匿名登录的话，也可以选择“匿名”。

（6）在用户文本框中输入 FTP 的用户名，在密码栏中输入 FTP 的密码。如果第 5 步选择了匿名的话，就不用输入用户名与密码了。

（7）单击“连接”按钮就可以连接到 FTP 站点。

在“站点管理”中，还可以设置传输模式、字符集等选项，在此就不再赘述了。

加入“站点管理器”中的 FTP 站点，会一直保存在 FileZilla 中。双击该站点名称，就可连

接到该站点。如果只是临时性地连接一个 FTP 站点，可以在“快速连接”栏中输入相关信息，进行连接，该站点会在 FileZilla 重启后消失。

4．在服务器与本机间传输文件

要将文件上传至 FTP 服务器是很简单的，只需要在图 20.1 所示“本地文件、目录”区中定位要上传的文件，然后将其拖动到右边“FTP 服务器文件、目录”即可。要下载 FTP 服务器上的文件，就可以从右边“FTP 服务器文件、目录”将文件或目录拖动到本地目录即可。

如果上传的文件与下载的文件出现了重名，则会提示如何处理这些重名文件，可以选择覆盖、重命名之类的操作。这个选项也是比较体贴的，可以防止错误地修改已存在的文件。

5．比较目录

如果要同步两个目录，就应该先比较两个目录中文件的内容。FileZilla 同样提供了比较目录的功能。分别在本地目录与 FTP 服务器目录中选择适当的目录，然后单击工具栏“切换为目录比较”按钮。可以看到两侧目录中的文件内容发生了变化，如图 20.5 所示。

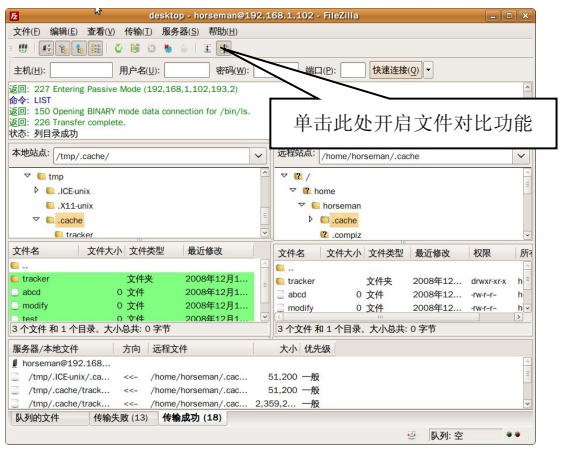


图 20.5　文件对比模式

20.3　字符界面软件 lftp

lftp 是一款命令行方式的 FTP 登录软件，其提供了很多强大的功能，相比图形界面的 FTP 登录软件功能是有过之而无不及。lftp 在 Linux 系统中也算得上是元老级软件了，但其并不因为年迈而失去生命力。实际上，到现在，lftp 在高级用户群体中使用还是很广泛的。而且，也不要觉得命令行操作的软件一定不好用或者一定是专家使用的东西。实际上，lftp 的常用命令是比较少的。而且这些命令一般也是一个简单的单词，其实记住也毫不困难。熟练记住了这几个命令后，大多数用户会爱上这款软件的。

20.3.1　FTP 登录

登录就是使用户名与密码输入到一个 FTP 服务器，登录后就可与其进行传送文件等操作。使用 lftp 登录 FTP 站点，可以使用如下命令：

```
lftp 用户名:密码@ftp 地址:传送端口（如果不设置，则默认为 21）
```

如使用“lftp ftpuser:234dfe@ftp.ubuntu.org.cn”，意思是以用户 ftpuser 登录 FTP 站点 ftp.ubuntu.com，用户密码是 234dfe。

由于命令行中的所有输入均会被保存到 Shell 历史记录中，所以使用该命令会很容易让其他人看到用户的 FTP 站点账号与密码。为了安全起见，可使用另一个命令来开始 FTP 站点登录，如下所示：

```
lftp FTP站点地址
```

由于不将密码在 Shell 中输入，可以保护密码的安全。下面是一个使用后一种方法进行登录的示例，步骤如下。

（1）使用“lftp 192.168.1.102”开始登录：

```
earthfans@mylaptop:~$ lftp 192.168.1.102   #登录到FTP站点192.168.1.102
```

（2）在 lftp 命令提示符中使用命令 login 开始以用户名 westnorth 登录，使用如下命令：

```
lftp 192.168.1.102:~> login westnorth    #使用login命令登录，用户名为westnorth
口令:                                    #输入westnorth的密码，输入并不回显在终端。
```

（3）正确输入用户名与密码后，就可登录到 FTP 服务器，执行相应的命令。本例执行 ls 命令：

```
lftp horseman@192.168.1.102:~> ls
#使用ls命令测试连接是否成功，如果出现文件列表信息，说明已经成功登录到FTP服务器。
total 232
-rw-r--r--  1 horseman horseman   234 Dec  9 05:19 #.newsrc-dribble#
-rw-------  1 horseman horseman  1910 Dec 16 02:41 .ICEauthority
-rw-------  1 horseman horseman   217 Dec 16 02:41 .Xauthority
-rw-------  1 horseman horseman  3562 Dec 14 10:54 .bash_history
-rw-r--r--  1 horseman horseman   220 Dec  8 17:38 .bash_logout
-rw-r--r--  1 horseman horseman  3115 Dec  8 17:38 .bashrc
```

还可使用如下方式进行登录。

- 如果 FTP 服务器支持匿名登录的话，则可使用一种特殊的账号 anonymous 进行登录。对于该服务器，登录用户名是 anonymous，密码可以是任何合法的电子邮件地址（格式为 xxx@xxx.xx）。
- 在 lftp 环境中，还可使用命令“open [-u user[，pass]] [-p port] host|url”进行登录。如“open -u westnorth，123456 192.168.1.101”。让用户 westnorth 登录到 192.168.1.101，用户密码是 123456。
- 还可以在 lftp 命令提示符中使用 open 命令进行登录：

```
open 192.168.1.101
user westnorth 123456
```

这样就可使用户“westnorth”，密码“123456”登录到“192.168.1.101”站点。

20.3.2　FTP 服务器目录与文件操作

连接到 FTP 服务器后，进入的 FTP 服务器目录不一定就是要下载文件所在的目录。这时，就可以使用一些命令来更改当前目录。

1．更改 FTP 服务器目录

在 lftp 命令提示符下，可以 cd 命令切换到想要下载文件的目录。如果不知道当前目录是什么，可以使用 pwd 来查看，然后使用 ls 命令查看当前目录中有哪些子目录，就可以使用 cd 切换到该子目录。这一切，与在终端中操作本地目录没什么区别。

一般情况下，FTP 站点在目录中会给出一个说明性的文件，该文件会简要介绍目录中文件的版本等信息。下面是一些显示文件的命令示例：

```
lftp horseman@192.168.1.102:~>pwd          #显示当前目录
/home/westnorth/
lftp horseman@192.168.1.102:~> ls          #显示当前目录中的文件
total 232
-rw-------  1 horseman horseman  1910 Dec 16 02:41 ICE
-rw-------  1 horseman horseman   217 Dec 16 02:41 authority
-rw-------  1 horseman horseman  3562 Dec 14 10:54 history
drwxr-xr-x  3 horseman horseman  4096 Dec 16 06:58 cache
drwx------  3 horseman horseman  4096 Dec  8 18:00 compiz
-rw-rw-rw-  1 hosreman horseman  2333 Dec  1 14:23 readme
lftp horseman@192.168.1.102:~>cd cache     #改变当前目录到其中一个目录
lftp horseman@192.168.1.102:~>cd ..        #回到上级目录
lftp horseman@192.168.1.102:~>cat readme   #显示 readme 文件的内容
this is the my FTP server.
this file contain test information.
```

2．操作 FTP 服务器的文件与目录

有许多时候，需要删除 FTP 服务器的文件与目录，或者需要新建目录之类。这些操作也与本机上的终端命令保持了一致性。例如：

- rm：删除文件；
- mrm：批量删除文件；
- mkdir：建立目录；
- rmdir：删除目录；
- chmod：改变文件、目录的属性；
- mv：移动文件。或者说是重命令功能。

3．在 FTP 服务器上查找文件

可以使用 find 命令来查找 FTP 服务器上的文件，如下示例是寻找后缀为 tar.gz 的所有压缩文件：

```
find . -name "*.tar.gz"
```

20.3.3 下载、上传文件

lftp 提供了丰富的下载、上传文件的命令。可以完成单个文件、多个文件的传输，甚至还可以目录为单位上传与下载。下面是一些简单的介绍。

1．多文件下载命令 mget

可以使用简单的正则表达式来匹配多个文件以供下载。mget 可取如下的参数。

- -c：继续，使用此选项文件传输中断后，下次传输可以接着本次传输的内容进行。
- -d：建立与下载文件同名的目录，将文件放置于该目录。
- -e：传输完后删除远程文件。
- -a：使用 ASCII 模式，默认是 Bin（二进制格式）。
- -O <base>：指定下载文件的存储位置。

用法示例：

```
mget -c -O ftpsoftware/ Lftp*
#下载服务器中以 Lftp 开头的所有文件，将其保存到 ftpsoftware 目录。并且开启断点续传功能。
mget -d Lftp18.zip
#下载 Lftp18.zip 文件，并将其保存到 Lftp18 目录中。
```

2．多线程下载命令 pget

多线程下载，可以加快下载速度。与 mget 一样可以使用“-c”、“-O”参数来指定断点续传与指定本地目录，除此之外，还可以使用“-n”参数指定下载的线程数。

```
如:pget -n 10 hugefile.tar.gz
```

3．上传文件命令 put

上传一个文件到 FTP 服务器可以使用命令 put。put 可取如下的参数。

- -o：指定 FTP 服务器的目录。
- -c：断点续传。
- -E：上传成功后删除本机的文件。

put 还有“-a”、“-O”等参数，意义与 mget 相同，这里就不再详细说明了。

4．多文件上传 mput

mput 可使用简单的正则表达式匹配多个文件供上传。除了功能与 mpget 相反外（mget 是下载文件，mput 是上传文件），mput 的可选参数与 mget 完全相同。

20.3.4　镜像命令 mirror

mirror 是一个功能强大的命令。可以将远程站点的目录镜像到本地、也可将本地目录及文件上传到服务器、还支持比较文件同步（比较文件的创建、修改时间，然后自动下载或上传新的文件，而其余文件不变）。其用法如下：

```
用法：mirror [选项] [远程站点目录 [本机目录]]
```

可用选项如下：

- -c：继续一个镜像任务。
- -e：删除本机存在，而服务器上不存在的文件。
- -n：将两个同名文件进行比较，然后只下载新的文件。
- -r：不下载子目录。注意这个与其他命令的用法刚好相反，使用 r 参数时是不递归处理子目录。
- -R：将文件上传至 FTP 服务器，这在上传文件时很有用，一般在使用时会伴随着-n 参

数，意思是只上传新文件。当使用该参数时，第一个目录变成了本地目录，而第二个成为远程目录。可以这样理解：在 lftp 中，将发送文件的计算机称为本地，将接收文件的计算机称为远程目录，这样就不会在反向镜像时弄错了。

- -N file：只下载比 file 新的文件。
- -i mat：只包括匹配的文件。
- -x mat：不包括匹配的文件。
- --log=file：将执行过的命令写入 file。
- --script=file：把命令写入脚本文件，但不执行。

由于 mirror 命令在使用时很容易出错，所以稍微详细地介绍一下该命令。假设本机的目录结构与 FTP 站点的目录结构如图 20.6 所示。本机的当前目录是“Local”，而 FTP 服务器的当前目录是“top”。

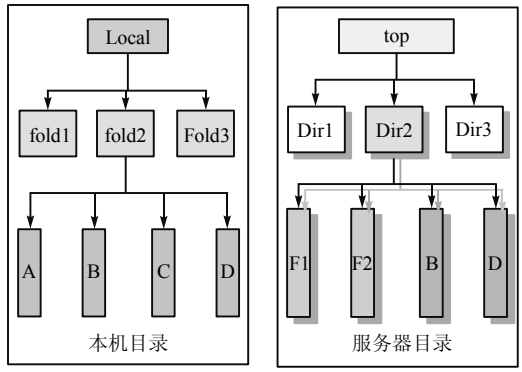


图 20.6　本机与远程 FTP 站点的目录结构

如果要同步本机的当前目录（本例为 local）与远程 FTP 站点的当前目录（本例为 top），则简单地使用没有参数的 mirror 即可：

```
mirror -n
```

如果要将服务器的 fold2 目录镜像到本地，应该使用如下命令：

```
mirror -n top/fold2 local/
```

镜像完成后，本机的 fold2 目录中会多出 F1、F2 两个文件。同时，如果本机的 B、D 文件修改时间晚于 FTP 服务器中相应文件，也会被替换为较新的文件。

而如果要将本机的 fold2 目录镜像到服务器，应该使用如下命令：

```
mirror -Rn local/fold2 top
```

镜像完成后，FTP 服务器会多出 A、C 两个文件。同时，FTP 服务器中的 B、D 文件也会被更新为最新的文件（如果其本身已经是新的，则保持不变）。

20.3.5　lftp 使用技巧

通过前面内容的介绍，应该可以完成相应的 FTP 操作。但 lftp 还有一些相应的技巧，掌握这些技巧，可以提高工作效率。

1．执行本机的命令

在执行 lftp 命令的时候，还可同时执行本机的命令。只要简单地在命令前加上“!”即可。如“!mkdir download”：在本机上建立一个名为 download 的目录。

2．模式设置

在前文曾经提及，FTP 在传输时有两种模式，分别是主动模式与被动模式。这两种模式分别有其不同的用途。在 lftp 中也可设置这些模式来满足不同的需求。设置的命令分别是：

```
set ftp:passive-mode 1      #使用被动模式登录
set ftp:passive-mode 0      #设置为主动模式
```

3．编码设置

在使用 lftp 登录中文站点时存在着编码问题。因为 lftp 默认的编码格式是 UTF-8，如果站点编码是 GBK 则会出现编码错误，而显示出乱码。这时可用 set 命令设置编码。将编码设置为 GBK 的命令如下：

```
 set ftp:charset gbk
```

如果要将编码设置为 UTF-8，只需要使用以下命令即可：

```
set file:charset utf8
```

4．使用书签

如果有几个经常登录的 FTP 站点，可以将其加入书签，以便使用。书签的使用很简单，简单使用命令 bookmark 即可，如下所示则将当前登录的 FTP 加入名为 uftp 的书签：

```
bookmark add uftp
```

然后在登录时，直接使用“lftp uftp”即可。bookmark 还有如下一些参数。

- del name：删除指定的书签。
- edit：编辑书签文件。
- list：列出书签，如果直接使用 bookmark 不带任何参数，则相当于使用了 list 参数。

5．使用别名

如果有些比较长的命令会经常用到的话，可以将其加入别名表。如：

```
alias gbk  set ftp:charset gbk          #给 set ftp:charset gbk 取别名为 gbk
alias syn mirror -Rn myweb/content ～/myweb # mirror -Rn myweb/content ～/myweb 的别名是 syn
```

然后就可直接在 lftp 中使用 GBK 来转换编码，用 syn 来同步内容。

6．退出连接

如果在 FTP 服务器的操作已经完成，就可以使用 close 来关闭连接。



字符界面软件与图形界面软件各有优劣。字符界面软件相对来说速度比较快、操作灵活程度高、可完成许多复杂的工作、或者可以快速完成重复的工作。图形界面的软件相对来说操作容易上手、比较直观。Linux 世界中有许多高效的字符界面软件，由于其高效灵活，是高级用户首选的软件。

20.4 WU-FTP 的设置

前面介绍了登录到 FTP 服务器的各种操作方法。本节介绍 FTP 服务器设置的相关知识，主要介绍了几个重要的配置文件的意义。

1. 重要的配置文件 ftpaccess

WU-FTP 服务器最重要的配置文件是/etc/wu-ftpd/ftpaccess，该文件可以配置的内容包括用户管理，以及用户权限等选项。该配置文件内容可以用任何文本查看软件，如“cat /etc/wu-ftpd/ftpaccess”查看。其内容如下所示（前面有两个#的为笔者所加的注释）：

```
    # Debian default wu-ftpd 'ftpaccess' configuration file, derived from
    # the 'ftpaccess.heavy' example in wu-ftpd sources.
    # For more options/commands see ftpaccess(5) and /usr/share/doc/wu-ftpd/*.
    # Some of the example message files have been translated to Spanish,
    # and are available in /usr/share/doc/wu-ftpd/examples/.
    # (thanks to Javier Fernandez-Sanguino Pen~a <jfs@dat.etsit.upm.es>
    # E-mail address of the FTP admin, can be accessed via the %E in messages.
    email ftpadmin@misconfigured.host
    ##这是系统管理员的邮件地址，可以对其内容进行修改。也可以在其他设置文件中使用“%E”调用该值。
    # Which UIDs and GIDs may, and which may not use the FTP service.
    deny-uid %-99
    deny-gid %-99
    allow-uid ftp ftpadmin
    allow-gid ftp ftpadmin
    ##设置哪些用户可以登录，哪些用户不能登录。先用“deny-uid”与“deny-gid”禁止所有的用户与组登录，然后使用“allow”命令允许某些用户登录。这在安全设置中，是常见的做法。
    # Maximum number of retries after login failures, before disconnecting.
    loginfails 5
    ##允许密码输入错误次数，如果设置为 5。密码输入错误 5 次，则会中止连接。
    # Can users see anything else but their home directory
    restricted-uid lamer
    unrestricted-gid ftpadmin
    ##是否允许用户访问其自己主目录外的内容。本例设置是允许用户 ftpadmin 访问，而不允许用户 lamer 访问。
    # Allow use of private file for SITE GROUP and SITE GPASS?
    private no
    ##用户登录后，是否可以成为一个增强组的成员，拥有该组的权限。设置为“no”是不允许的。
    # What kind of greeting to give.
    greeting <full|brief|terse|text somemessage>
    ##用户登录后，显示的欢迎信息内容。
    banner /etc/wu-ftpd/welcome.ms
    ##在用户登录前，显示的说明信息。
    # Deny access to specified hosts, with message.
    deny    *.microsoft.com         /etc/wu-ftpd/msg.deny
```

```
deny    /etc/wu-ftpd/denied.hosts   /etc/wu-ftpd/msg.deny
```

##拒绝特定的域名或 IP 地址的用户访问该 FTP 站点，并显示提示信息。/etc/wu-ftpd/msg.deny 就是提示信息的内容。

```
# !nameserved means hosts that can't be resolved.
deny    !nameserved /etc/wu-ftpd/msg.nodns
```

##如果使用“!nameserved”则意思是如果域名无法解析，则拒绝访问。

```
# Various DNS related options.
#dns refuse_mismatch <filename> [override]
#dns refuse_no_reverse <filename> [override]
#dns resolveroptions [options]
# Class name    typelist        addresses
class   local   real,guest,anonymous    *.earthfans.cn 192.168.1.*
class      remote    real,guest      *.google.com
class    all real             *
```

##class 命令用来定义类，这些类拥有同样的权限及特性，可在后面的设置中用到。name 列是类名。typelist 是 FTP 用户类型。FTP 的用户可分为三类，real 是在服务器上拥有登录账号与密码的用户，guest 是可以登录 FTP 的账号，而 anonymous 是匿名用户。address 列是该案类的地址。可以使用通配符，同时可使用域名与 IP 地址。

##本例中 local 类是指 earthfans.cn 的二级域名，以及 192.168.1.1-192.168.1.255 的所有 IP 地址，包含所有类型的 FTP 用户。remote 类包含 google.com 的二级域名，包含了 real 与 guest 类的用户。

```
# Limit who how many    date/time   message file
limit   local   20  Any                 /etc/wu-ftpd/msg.toomany
limit   remote  100 SaSu|Any1800-0600 /etc/wu-ftpd/msg.toomany
limit   all     10      Any             /etc/wu-ftpd/msg.toomany
```

##limit 限制某类用户的使用限制。如第二行，第二列是类名，意思是限制的范围是 remote 类成员。第三列是最多允许连接的数量，如 100，则最多只允许同时有 100 人登录。第四列是登录时间，“SaSu”表示只允许周六周日访问。这两个词是星期的缩写。“|Any1800-0600”的意思是，可以在任何一天的 18 点到第二天早 6 点之间访问该服务器。最后一列是如果超出此限制试图连接到服务器得到的错误提示信息。

```
# The files that wu-ftpd will recognize as must-be-read, and display them.
message /welcome.msg          login
message .message              cwd=*
```

##message 是用户执行特定的指令时，系统显示的特定内容。第一行的内容是用户登录时显示 welcome.msg 的内容。第二行的意思是用户更改目录时，显示.message 文件内容。

```
readme  README*    login
readme  README*    cwd=*
```

##readme 命令的意思是用户执行特定的操作时，显示目录中的 README 文件。

```
compress        yes           local remote all
```

##指定哪些用户可以使用 compress。yes 为可以使用，而 no 则为不可使用。

```
tar           yes           local remote all
```

##指定哪些用户可以使用 tar。

```
log transfers anonymous,guest,real inbound,outboun
```

##指定哪些用户的操作会被记录到日志文件中。

```
shutdown /etc/wu-ftpd/shutmsg
```

##从“/etc/wu-ftpd/shutmsg”文件读取关闭 FTP 服务器的时间，时间到则无法进行登录。关闭 FTP 的文件可由 ftpshut 建立。

```
# These files are marked unretrievable
noretrieve /etc/passwd /etc/group
noretrieve core
```

##设置不允许访问的文件类型第一行指定任何用户不可访问/etc/passwd 与/etc/group，第二行指定任何用户均不允许访问所有名为“core”的文件。

```
anonymous-root /home/ftp
```

##设定匿名用户的主目录。

```
passwd-check    rfc822  enforce
```

##设置匿名用户密码验证格式。第二列为密码格式，none 为不做密码验证，trival 只要输入密码中含有“@”字符就可以登录，而 rfc822 是最为严厉的，要求必须符合（***@***.**）之类的格式才可以通过验证。第三列是验证方式，如果为 enforce，则密码不符合格式规定，就无法登录。而 warn 则表示密码不符合格式，也可以登录，不过会显示一条警告消息。

```
limit-time anonymous 30
```

##限制特定用户所允许的持续登录时间，单位是分钟。默认是无限制。

```
rename      no  anonymous       # rename permission?
delete      no  anonymous       # delete permission?
overwrite   no  anonymous       # overwrite permission?
chmod       no  anonymous       # chmod permission?
umask       no  anonymous       # umask permission?
```

##用户执行某种命令的权限。本例中，使 anonymous 用户没有 rename，delete，chmod 等权限。

```
#   anon-ftp-root   path        allow?  owner   group   mode    dirs?
upload  /home/ftp   *       no
upload  /home/ftp   /pub/incoming   yes ftp daemon  0666 nodirs
```

##指定用户的上传选项。第一列指定了用户的根目录，该目录与/etc/passwd 文件中定义的用户主目录应该是一致的，本例中是“/home/ftp”。第二列指定了用户可上传的目录，如“/pub/incoming”。第三列是否允许上传，yes 为允许上传。第四列为文件上传后的属主，如“ftp”意为文件上传到“/pub/incoming”目录后，文件会属于用户“ftp”。第五列为文件上传后的属组。第六列为文件的属性，如是 666 则为所有用户可读、可写。第七列为是否允许在用户主目录中创建目录，nodir 为不允许建立目录。在 upload 命令中，还可使用 class=cname 来指定限定的用户类。

```
path-filter anonymous   /etc/pathmsg  ^[-+A-Za-z0-9_.]*$  ^\.  ^-
```

##指定特定用户上传文件所允许使用的字符

```
alias   incoming:   /pub/incoming
```

##定义目录的别名，可以在 FTP 中使用。如上例中定义的目录别名，可以使用 cd incoming 来切换到“/pub/incoming”目录。而且，在定义时，必须使用“:”。

```
cdpath  /pub
```

##增加系统路径。将/pub 添加到系统路径中。也可以添加其他目录到系统路径中。

2．其他配置文件

除了 ftpaccess 外，lftp 的还有其他几个重要配置文件，将其位置及功能列出如下。

- /etc/wu-ftpd/ftpusers：禁止登录 FTP 服务器的用户名，每行一个用户名，表示不能进行登录的用户。
- /etc/wu-ftpd/ftpconversions：在 FTP 传输过程中对文件进行压缩。
- /etc/wu-ftpd/ftphosts：禁止指定机器登录，文件格式类似如下所示：

```
deny    myguest    192.168.1.*
allow   westnorth  192.168.1.*
```

是允许用户 westnorth 从 192.168.1.*网段访问，而拒绝用户 mygues 从该网段访问。

3．一个重要的脚本的修改

现在，修改了 ftpaccess，读者可能会满怀信心地重启 WU-FTP 服务，然后期待着新的设置能够生效。但遗憾的是对于 ftpaccess 的设置似乎完全没有效果。不过这不是 bug，也不是 Ubuntu 系统的问题，这只是一个安全设置。其关键就在 WU-FTP 的配置文件“/etc/init.t/wu-ftpd”中。下面是修改的具体步骤。

（1）用 root 身份打开该文件，如果不知道如何使用 root 身份打开的话，可以使用“sudo gedit /etc/init.t/wu-ftpd”命令。

（2）在该文件大约第 28 行处找到“WU_OPTIONS="-l"”，在这里可以设置 wu-ftpd 的启动参数。默认设置的参数“l”的意思是将 lftp 的日志写入系统日志文件中。参数“-a”的功能是使其在启动时读取 ftpaccess 配置文件，而 WU-FTP 是默认不读取 ftpaccess 的。所以只需在参数中加上“a”参数即可。修改后的语句应该是这样：

```
WU_OPTIONS="-la"
```

（3）使用“sudo /etc/init.d/wu-ftpd restart”重启 FTP 服务即可使得 ftpaccess 中的设置生效，也可以进行匿名登录了。



WU-FTP 的优势在于其丰富的可设置性，这些特性使得 WU-FTP 更为强大、灵活与安全。但随之而来的就是这些复杂的设置将许多初学者挡在了门外，读者可按照自己的需要灵活选择。

20.5　添加只能进行 FTP 操作的账号

在对 ftpaccess 进行设置后，就可以开放自己的 FTP 服务器供其他用户下载了。但为了对登录的用户（非匿名用户）进行管理，还需要在系统中添加进行 FTP 操作的账号。进行管理的最好方法就是设置一些只能进行 FTP 操作的账号。添加只能进行 FTP 的账号也是比较容易的，下面是添加用户的步骤。

（1）先添加一个普通用户，使用如下命令：

```
sudo adduser jack
```

然后按照提示输入相关信息，如密码、个人信息等。命令完成后，应该已经创建了一个叫 jack 的用户。

（2）一般情况，进行 FTP 登录的账号是不应该登录进系统的。所以还应该修改其登录 Shell。使其不能登录进入系统，使用如下命令将 jack 的登录 Shell 修改为 false：

```
sudo usermod -s /bin/false jack
```

现在，就创建了一个只能进行 FTP 登录的账号。如果要用该账户登录系统，会简单地得到一个错误。

（3）但仅仅修改了登录 Shell 的账号还是有些安全问题的，如该用户可切换至一切目录，查看系统文件等。所以要修改/etc/wu-ftpd/ftpaccess 文件来限制其权限。该文件中关于用户权限的修改有如下几项内容：

```
    email westnorth@gmail.com
    #修改管理员的密码
    deny-uid %-99 %65534-
    deny-gid %-99 %65534-
    #由于正常的用户及组 ID 均在 100 到 65534 之间，超出此范围的，可以视为对系统的攻击，将其拒绝。
    allow-uid ftp
    allow-gid ftp
    loginfails 2
    #密码错误次数，达到该次数后，会自动断开连接。这也是为了防止有人暴力破解密码。
    restricted-uid jack
    #将用户 jack 限制在其主目录中。其只能查看，修改其主目录中的文件。
```

如果还有其他需要，可以参照前面的介绍进行修改。



添加这些只能进行 FTP 操作的账号有其显而易见的好处，就是易于进行管理。同时，由于这些账号不能用于系统登录，也杜绝了一些安全隐患。这些账号只存在于配置文件中，不占用系统的其他资源（因为其没有专用的主目录），对于作为 FTP 服务器的主机来说也减轻了一定的负担。

20.6 小结

本节介绍了互联网世界中一种非常有用的服务 FTP 服务的相关知识，包括 FTP 服务器的架设、登录 FTP 服务器的方法等。本章的难点内容是 WU-FTP 的设置与添加 FTP 账号。但这两项内容在实际的应用中却经常用到，所以读者应该多加分析 WU-FTP 的配置文件，理解其各项的意义，同时练习添加只能 FTP 操作的账号。对于普通用户而言，登录 FTP 的机会可能更多些。lftp 功能强大，使用又比较简单，应该多练习 20.3 节中的命令，熟练掌握这个软件的使用方法。

第 五 篇

Linux 安全篇

- 第 21 章　用户与组管理
- 第 22 章　系统安全

第 21 章　用户与组管理

Linux 是一个真正的多用户操作系统，不同的用户对于同一目录文件可能拥有有完全不同的权限。用户可以对属于自己的文件设置一定的权限，使得其他用户无法读取、写入或运行。而系统管理员又可以给不同的用户赋予不同的权限，控制用户消耗的系统资源。所以，对于用户的管理是系统管理中的一个重要组成部分，也是普通用户所必须具备的基本技能。本章就来介绍用户与组管理的相关知识。

21.1　用户账号的基本操作

用户账号在 Linux 系统中的作用是标识用户身份，正确使用用户名和密码才能登录进系统。然后系统就会给用户分配用户空间，该目录被称为用户的主目录。在终端中，用“～”来指代用户的主目录。用户在自己的主目录中拥有最大的权限，可以进行所有的文件操作。本节就介绍用户账号的建立、删除与修改的基本操作。

21.1.1　用户账号类型

Linux 系统的账户是用户用来登录系统的唯一身份标识，系统中有三种类型的账号，每种账号有其特殊的权限设定。下面分别介绍这三种账号。

1．超级用户 root

Linux 系统在安装后，默认存在着一个超级用户 root。该用户拥有所有的权限，可以在整个系统进行所有的操作。但由于其权限过高，稍有不慎会造成系统核心配置文件的修改，可能会造成严重的后果。而且如果系统被黑客入侵，就会获得最高权限，对于系统安全来说也有很大的风险。由于平时的操作也不需要太高的权限，所以，登录时应尽量避免使用 root 账号登录。root 用户不能删除，也不能新建，系统中只能有一个 root 用户。

2．虚拟账号

除了超级用户 root 外，系统还存在着一种称为虚拟账号的特殊账号。这些账号并不是用来提供给用户登录用的，而是为来执行某些特殊操作的，如 FTP、E-mail 等。由于其不能进行实际的登录，所以这些账号有时被称为伪账号。

3．普通账号

还有一种是普通用户经常会接触到的，就是普通账号。这种账号相比超级用户 root 少了许多权限。而相对于虚拟账号而言，这些账号又多了登录系统的功能。一般而言，用户登录使用的账号均属于普通账号。



Linux 设置了这三种类型的账号，是有其原因的。root 用户用于系统管理，虚拟账号用于特殊的操作，普通账号则是用于普通用户日常操作。这样的设置，相对于用户种类少的操作系统，多了许多灵活性。

21.1.2　建立用户账号

建立用户账号就是在系统中增加一个账号，同时还为该账号分配相应的其他资源，如以数字形式表示的账号 ID、该账号所属的组、用户的主目录、登录系统后使用的 Shell 等。而且，要想让该账号投入使用，还需要给账号指定一个密码，否则该账号同样是无法登录的。下面是建立用户账号的相关知识。

1．useradd 命令简介

建立账号可使用命令 useradd。其最简单的使用方法是直接在命令后跟账户名，如“useradd guest”。该命令会进行下列操作。

（1）读取用户配置文件/etc/login.defs。

（2）读取用户配置文件/etc/default/useradd。

（3）根据以上两个文件定义的规则添加用户。

（4）向/etc/passwd 和/etc/groups 文件添加用户和用户组记录。

（5）系统还会自动在/etc/add/default 中所约定的目录中（一般情况是/home 目录中以账户为名的一个目录）建立用户的主目录。

（6）系统会复制/etc/skel 中的文件（一般是些系统设置文件，如.bashrc 等，还有一个示例目录 Example）到新用户的主目录中。

useradd 的完整语法格式如下：

```
useradd option username
```

其中 option 可以取如下的值。

- -c comment：给该用户设置一点说明信息。如“this is a cat”。
- -d dir：指定用户主目录。如果目录不存在，则可同时使用-m 参数就可在创建用户的同时创建主目录。
- -e DATE：指定账号终止日期，日期的指定格式是：MM/DD/YY。
- -f n：当 n 为 0 时，如果账号终止日期到，则账号立即被停止。默认值是-1，意思是不停止该账号。
- -g group：指定新建用户所属的用户组。
- -G group：指定新建用户所属的附加组。
- -k dir：默认创建用户主目录时，是从/etc/skel 目录复制文件。可以使用-k 参数更改目录。
- -m：用来在创建用户的同时，自动建立用户的主目录。
- -r：建立系统账号。该账号一般用于非登录用途，不会建立用户目录。
- -s Shell：指定用户的登录 Shell。
- -u ID：指定用户的以数字形式表示的用户号。最好不要与其他用户重复。一般从到 1000 开始，小于 65534。
- -D：特殊的参数，如果其后不跟参数，则显示当前系统的默认值。包括目录位置，组编号，登录 shell 等。也可以跟某些参数来更改相应的系统默认设置。

“username”用来指定新建账号的用户名。

2．username 使用实例

下列命令创建一个名为 guest 的用户。如果/home/guest 目录不存在的话，会自动创建该目录（由-m 参数指定）。命令如下：

```
sudo useradd -d /home/guest -m guest
```



即使不使用“-d”参数显式指定用户主目录的话，系统也会自动将该用户的主目录设置为“/home/guest”。

创建一个名为 temp1 的临时用户，用如下命令：

```
sudo useradd -s /bin/sh -e 12/25/08 -g guest -G temp,user -m temp1
```



之所以称其为临时文件，是因为该账号会于 08 年 12 月 25 日过期（-e 参数指定）。该用户使用 sh 作为登录 shell，属于 guest 组，而且附属于 temp 与 user 组。

创建一个名为 sysuser 的系统用户，该用户没有主目录，不能登录但可用于其他用途，如 ftp、ssh 等连接中。同时显式指定该用户的 UID 为 1202，命令如下：

```
sudo useradd -r -u 1202 sysuser
```

3．新建账号的交互式命令 adduser

adduser 与 useradd 命令功能甚为相似，然其操作是交互式的，可能有的用户更喜欢这种形式的操作。下面是一个建立用户的实际过程。在命令行中输入“sudo adduser guest”即开始了交互式创建用户的过程，如下所示（以#开始的是注释行，其余均为命令的输出）：

```
sudo adduser guest  #该命令的使用很简单，直接在 adduser 的后面跟要的账号名即可。
正在添加用户 guest... #第一步是添加用户
正在添加新组 `guest' (1003)...
#建立一个与用户同名的组，组的 ID 是 1003，这也是 useradd 的默认行为，如果 useradd 不指定组的话，也将建立一个与用户的组。
正在添加新用户 'guest' (1002) 到组‘guest’...
#将用户添加到组中，用户 ID 为 1002。
创建主目录 '/home/guest'...   #创建与用户同名的用户主目录
正在从 /etc/skel 复制文件...
#默认行为是从/etc/skel 目录中复制一些必须的文件，如.bashrc，.profile 等。
输入新的 UNIX 口令：              #设置用户登录的密码
重新输入新的 UNIX 口令：
passwd：已成功更新密码
正在改变 guest 的用户信息
请输入新值，或直接敲回车键以使用默认值
        全名 []:
        房间号码 []:
        工作电话 []:
        家庭电话 []:
        其他 []:
这个信息是否正确？ [Y/n]y
#设置用户相关信息，如果显示的消息正确，则按“y”键完成创建用户的任务。
```

21.1.3　批量添加用户

从前面的介绍来看，向系统中增加用户是件比较容易的事。但是如果你是位大型网络的管理员或者简单地说是个学校的计算机教师，则会经常需要一次添加多个用户。如果还是使用 useradd 或 adduser 命令来逐个建立的话，那将是一件极其耗费时间的痛苦过程。而在很多情况下，也不可能花太多的时间用来建立用户。不过，幸运的是，Linux 提供了一种快速添加大量用户的方法来解决这个难题。下面就来介绍批量添加用户名的具体步骤。

（1）编写与/etc/passwd 文件格式相同的用户名列表文件。/etc/passwd 文件格式本章后面会进行详细的介绍，这里先给出一个示例文件，设其名为 userlist，内容应该类似于这样：

```
name1::1500:100:user:/home/name1:/bin/bash
name2::1601:100:user:/home/name2:/bin/bash
name3::1602:100:user:/home/name3:/bin/bash
name4::1603:100:user:/home/name4:/bin/bash
name5::1604:100:user:/home/name5:/bin/bash
name6::1605:100:user:/home/name6:/bin/bash
```

（2）将上一步建立的用户列表文件加入到/etc/passwd。注意，这里不用逐个手工添加，而应该使用命令 newuser，其使用方法应该类似这样：

```
sudo newusers < userlist
```

这样，就可将 userlist 文件中的用户名列表加入到系统用户库文件中。如果对该命令的运行效果不太放心，可以使用“sudo vipw”或“sudo vi /etc/passwd”检查列表中的用户是否已经被成功添加。

（3）取消影子口令，使用如下命令：

```
pwunconv
```

该命令的功能是根据 passwd 与 shadow 文件生成密码，然后删除影子口令。这是为了方便下一步的密码转换。

（4）新建一个文本文件，设其名为“password”（该名字可任意取）。文件由多行文本组成，每行的格式是”用户名：密码“。但内容要与前面的用户列表相对应，原来 uselist 文件包含了 6 个用户，这里的用户名与密码也应该有六个，而且用户名应该与 uselist 文件中的名字完全相符。对于格式，密码文件有一个小小的要求，就是文件中不能有空行。密码文件类似这样：

```
name1:123456
name2:34348fd
name3:dcvfe
name4:344#dfe
name5:138E#D#
name6:143439
```

（5）创建用户密码使用 chpasswd，该命令 password 文件中密码送入 passwd 命令，经过编码后，将编码后的密码内容写入/etc/passwd 的密码项中，chpasswd 命令的使用类似这样：

```
chpasswd < password
```

同样，还可以使用 vipw 或 vi /etc/passwd 命令来查看是否已经将相关密码写入了用户行中。

（6）现在密码文件以明文方式存入于/etc/passwd 文件，由于该文件对于普通用户来说是可读的，存在一定的安全隐患。所以应该创建影子口令，也就是将密码通过加密以影子口令的形式存入只能由 root 用户读取的文件中。创建影子口令是以命令 pwconv 来实现，而该命令使用也很简单，不用加任何参数直接运行即可，类似这样：

```
pwconv
```

通过以上 6 步，就可以将大量的用户添加到系统中，而且用户使用了影子口令，增强了安全性。而需要做的，只是建立两个文本文件即可。可以看出，这种方式比通过 adduser 之类的命令逐个建立要方便快捷得多。

21.1.4 编辑用户账号属性

当用户的身份发生改变，就需要对其进行相应的操作，如需要提高或降低其权限、或者直接将用户删除，使其无法登录等。这些操作被称为编辑账号，编辑账号的操作包括删除账号与修改账号，下面分别予以简单的介绍。

1．删除账号

当一个临时账号过期，或者使用账号的用户离职后，应该将该账号删除。删除账号就是删除账号在/etc/passwd、/etc/shadow、/etc/group 等账号信息文件中的记录。一般情况还应该删除该用户的主目录。删除账号可以使用 userdel 命令，如果对系统文件有足够的了解，也可以直接删除相关信息文件中的记录（危险，不推荐）。userdel 命令的格式如下：

```
userdel option username
```

- option 很简单只有一个，就是“-r”，意思是删除用户时将用户的主目录一同删除。
- username 就是要删除的账号名。

如“userdel -r temp”删除用户 temp，同时删除该用户的主目录（一般是/home/temp）。

2．修改账号

建立的账号出现了细微的错误该怎么办，是否应该将账号删除，然后再重新建立呢？如果要删除再重建，就需要记录用户的各种信息，如密码、主目录、所属组、登录 Shell 等信息。稍有不慎，建立的用户就会与原始的记录不一致，造成许多问题。理想的做法应该是修改用户的部分记录信息，而不触动其他信息。Linux 提供的修改用户账号命令是 usermod，其格式如下：

```
usermod option username
```

option 可用的选项有-c、-d、-e、-f、-G、-g、 -G、-l、-s、 -u 等，这些选项的用法与 useradd 命令中的一致，分别是指定用户的注释、主目录，停用日期、是不停权、用户所属组、附属组等信息。相比 useradd 而言，usermod 增加了-l 参数，可使用该参数更改用户名。如“usermod -l newname”是使用 newname 作为该用户的新账号名。

username 是要修改信息的账号名。

下面是该命令的两个使用示例：

修改 tom 的登录 shell 为 csh，主目录为/home/tom，所属组为 guest。可用如下命令：

```
usermod -s /bin/csh -d /home/tom -g guest tom
```

将用户 tom 更名为 tom2，可用如下命令：

```
usermod -l tom2 tom
```

21.2　管理用户密码

用户添加成功后，如果没有给该账号指定密码，该用户是无法登录的。所以，要使该用户生效，就应给该用户指定密码。或者如果用户密码遗忘或被泄露，这些情况都应该修改用户的密码。Linux 中，指定或修改用户密码，应该使用 passwd 命令。该命令的使用格式是：

```
passwd option username
```

option 可取许多参数，不过常用的也就是如下几个。

- -l：锁定口令，使该用户无法登录。
- -u：口令解锁，恢复该用户的登录权。
- -d：删除口令，这样该用户在登录时就不用输入口令了。
- -f：强迫用户下次登录时修改口令。
- -m n：指定口令修改的最小天数，也就是说，一个口令只能使用 n 天，超过 n 天，则系统会强制用户更改密码。

username 是要修改密码的账户名。

如果不使用 option 参数，则默认是修改用户密码，如果再不使用 username 参数，则默认是修改当前用户的密码。如果 root 用户使用 passwd 命令，可以修改任何用户的密码。普通用户使用 passwd 命令，则只能修改自己的密码，而且在修改密码前要输入原始密码。下面是几个使用 passwd 命令的实例。

例 1．给新用户指定密码：

```
sudo passwd temp1          #要指定新用户的密码，应该使用 sudo 切换到 root 模式。
输入新的 UNIX 口令： 123456af
重新输入新的 UNIX 口令：123456af
passwd：已成功更新密码
#指定密码成功，temp1 用户就可使用 123456af 进行登录。注意该密码要稍微复杂些。
```

例 2．修改自己的密码：

```
passwd                              #直接使用 passwd 命令，则为修改当前用户的密码
Changing password for temp1.        #提示是修改用户 temp1 的密码
(current) UNIX password:            #输入当前密码
Enter new UNIX password:            #输入新密码
Retype new UNIX password:           #将新密码再输入一次
Bad: new password is too simple
#如果输入的密码过于简单，则会获得该提示，然后应该继续输入。
Enter new UNIX password:
Retype new UNIX password:
Password unchanged
#如果两次输入的密码不一致，则提示密码没有更改成功。
Enter new UNIX password:
```

```
Retype new UNIX password:
passwd: password updated successfully
#更改密码成功，则提示更新成功。
```

例 3．检验密码是否修改成功，可用 su 命令切换到该账号进行试验：

```
su temp1                                    #使用 su 命令切换到用户 temp1 进行登录
口令： 123456we
#如果口令输入正确，就会以 temp1 的身份进行各种操作，而且当前目录会被更改为用户 temp1 的主目录。
$ ls -a
#显示目录中的文件，这些文件是新建账户时从/etc/skel 目录中复制过来的。
.  ..  .bash_logout  .bashrc  Examples  .profile
$ pwd
#显示当前目录，可以看到当前目录是用户 temp1 的主目录
/home/temp1
```

21.3 组的管理

对于一个复杂的网络系统而言，权限的分配是一个复杂的过程。如开发部门应该对开发所需的技术资料拥有更高的权限，而市场部门应该对产品资料、客户资料等拥有更高的权限，人力资源部门应该对从事资料拥有更高的权限。对于这样的要求，如果对数据进行授权的工作还是以账户作为依据的话，那授权或者更改授权都会是一件漫长且容易出错的工作。在这种情况下，就应该以“组”为单位进行管理。

Linux 中的组是许多用户的一个集合，该组的用户对于特定的资源拥有相同的权限。这样会对用户的权限管理带来许多方便，如对于 Develop 组，可以将相关的技术资料设置为可读、可执行等权限。而对于其他组成员，则设置为不可读、不可执行等。通过上面的介绍，应该可以大致了解 Linux 中关于组的概念。而用户与组之间的关系，又有如下几种。

- 一对一：一个用户对应于一个组。系统默认在建立用户时会建立与账号同名的组，但这种方式对于管理来说没有什么好处。
- 多对一：多个用户对应于一个组。将多个用户放置于同一个组，以赋予其相同的权限，这是管理中常见的一种模式。
- 一对多：一个用户属于多个不同的组。为了某些需要，可以允许一个用户属于多个组，如对于公司的高级管理人员而言，其应该可以有充分的权限查看技术资料、市场信息、人事情况等，就可以将高级管理人员放置于多个组中，同时拥有多个组的权限。

这些关系可以参照图 21.1 来进行理解。

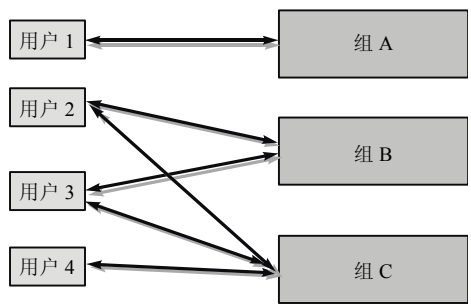


图 21.1　用户与组之间的关系

本节的内容是组的管理，包括添加组、删除组、更改当前组等内容。

1．添加组

添加用户组的命令是 groupadd，其使用格式是：

```
groupadd option groupname
```

option 是命令的参数，常用参数是：

-g：指定新用户组的 GID 号码。

groupname 是新用户组的名称。

如果没有使用-g 参数指定组号码，则会自动给最大的一个组号加 1，然后赋予新建组。用法如下所示。

新建一个名为 develop 的组，如果当前最大的组号为 120，则 develop 组号为 121。使用如下命令：

```
sudo groupadd develop
```

新建一个组，名为 manage，组号为 150。这里显式指定了组号，命令是：

```
sudo groupadd -g 150 manage
```

2．删除组

删除组使用的命令是 groupdel，使用方式很简捷，直接在命令后跟要删除的组名即可。

如下所示的命令会删除名为 manage 的组：

```
groupdel manage
```

但是，该组中如果有用户已经登录到系统中，则不能进行删除。

3．修改组属性

修改用户组的属性使用命令 groupmod，其使用格式是：

```
groupmod option name
```

option 是命令的参数，除了可以取-g 修改组标识号 GID 外，还可使用“-n newname”指定组的新名字，也就是说，可以更名组。用法示例如下。

将组 manage 的 GID 更改为 180。使用如下命令：

```
sudo groupmod -g 180 manage
```

将组 manage 的名称更改为 CEO。使用如下命令：

```
sudo groupmod -n CEO manage
```

4．切换组

由于用户与组可以是一对多的关系，也就是说，一个用户可以属于多个组。为了获得其他组的权限，应该切换到用户所属的另一个组。切换组可以使用命令 newgrp。如：

```
newgrp CEO
```

将当前用户的组切换到 CEO 组，但是当前用户必须是属于 CEO 组，才能切换成功。

5．设置组密码

gpasswd 用来向组中添加密码。一般情况，没必要给组设置密码，但在某些安全性要求比较高的情况或者是在比较复杂的模型中，可能要求组有密码。可以使用 gpasswd 来实现。

gpasswd 命令的使用格式是：

```
gpasswd option gname
```

option 可取如下的参数。

- -a user：添加 user 用户到组中。
- -d user：将 user 从组中删除。
- -r：去除组密码。
- -A user：设置 user 用户为组管理员。
- -M user1，user2..：将多个用户添加到组中。

gname 就是组名。

gpasswd 的一般用法如下：

```
gpasswd devel                          #修改组 devel 的密码
正在修改 devel 组的密码
新密码：                               #输入密码
请重新输入新密码：                     #重复输入密码
```

这样就完成了设置组密码的过程。

21.4 用户、组的配置文件

在创建用户与组的过程中，其实所有的相关信息均写入了相应的配置文件。如用户信息保存在“/etc/passwd”文件，而组信息则保存在“/etc/group”中。通过查看这些配置文件，就可以查看系统中用户与组的全部信息。而且高级用户还可对用户与组配置文件进行修改，以直接添加或删除用户、组。本节就详细介绍了用户、组相关的配置文件及其文件内容的意义。

21.4.1 账户信息配置文件/etc/passwd

在介绍账号管理相关知识时，提到了账户信息的配置文件/etc/passwd。因为在系统创建新的用户时，其实是向该文件里写入了相关的信息。本节就来详细分析一下该文件的内容。通过了解该文件与/etc/shadow 文件，就可以自行编辑这两个文件来创建用户了。一个典型的/etc/passwd 文件内容如下所示：

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
klog:x:102:103::/home/klog:/bin/false
messagebus:x:103:106::/var/run/dbus:/bin/false
westnorth:x:1000:1000:Laptop Ma,,,:/home/westnorth:/bin/bash
dictd:x:112:126::/var/lib/dictd:/bin/false
mldonkey:x:113:128::/var/lib/mldonkey:/bin/false
mysql:x:114:129:MySQL Server,,,:/var/lib/mysql:/bin/false
Debian-exim:x:115:130::/var/spool/exim4:/bin/false
fetchmail:x:116:65534::/var/lib/fetchmail:/bin/sh
```

passwd 文件由许多行记录组成，每行记录代表一个账号。而各行又由“:”分隔开的七列

组成。各列的含义分别是：

```
账号：密码：用户号(UID)：组号(GID)：账号的描述信息：该用户的主目录：登录 Shell
```

下面分别对各项内容进行详细解释。

- 账号：记录了系统中的所有账号。由于这是一个单机用户的 passwd 信息，其中除了 westnorth 外，其余账号均是系统账号，也就是无法登录的账号。但这些账号是系统中其他相关进程所需要的，不能进行修改与删除。如 fetchmail 用来接收信件，mysql 是 MySQL 数据库所需要的用户。
- 密码：记录了账号的密码。其实这里记录的也是经过加密的密码。但由于是明文表示的密码，而且该文件对于普通用户来说，也是可读的，所以为了安全起见，又在加密的密码基础上产生了一个影子口令，存放于/etc/shadow 中，这列的“x”表示有密码，但密码是影子口令。如果为空，则说明账号无密码。
- 用户号（UID）：在建立用户时，会自动分配一个十进制数字的 id 号，用来标识其身份。一般来说，该 ID 号应该是唯一的。但也可多个用户使用同一个 id 号。Ubuntu 的普通用户 ID 号是从 1000 开始递增的。而系统账号的 ID 是从 100 开始递增的。所以简单地通过查看 UID，就可以知道哪些账号是系统账号。如 UID 为 114 的账号为系统账号。
- 组号（GID）：为了标识用户组，系统也会给组分配一个十进制数字的 ID 号。Ubuntu 同样是从 1000 开始递增。同样地，系统账号的 GID 也从 100 开始递增。
- 账号的描述信息：一些关于该用户的描述信息。对于普通用户而言存放的信息是 finger 所显示的内容。对于系统账号，则显示其大概的功能。如 mysql 的描述信息是“MySQL Server”。
- 该用户的主目录：对于普通用户而言，主目录是拥有全部权限的“自己的地盘”，可以在主目录中进行任意操作。而对于系统账号而言，主目录是读取设置信息或是进行特定操作的目录。
- 登录 Shell：系统启动后，需启动一个用户与系统内核之间的中间层，用来传递用户的操作与系统的输出，这个中间层就是 Shell。虽然各个 Shell 有不同的特色，但其基本功能均是类似的。常见的 Shell 有 Csh、Ksh 等，但 Ubuntu 默认并未安装这些 Shell。所以对于 Ubuntu 而言，可用的 Shell 就只有 sh 与 bash 了。但是，登录 Shell 并不是必须指定为 Shell 程序，指定为其他特定的应用程序也可以。这样，该用户就会在运行了指定的程序后退出。如 sync 的登录 Shell 是/bin/sync，说明 sync 账户会在从开始的后执行 sync 命令。如果是/bin/false，执行/bin/false 程序。

21.4.2　账户的影子口令文件/etc/shadow

/etc/shadow 是为了增强密码的安全性，而由 pwconv 命令创建的影子口令文件。其内容与/etc/passwd 的内容相对应，也由一行行文本组成，文本同样由“：”隔开，各列分别代表不同的意义。一个典型的 shadow 文件内容类似这样：

```
root:!:14214:0:99999:7:::
daemon:*:14214:0:99999:7:::
bin:*:14214:0:99999:7:::
sys:*:14214:0:99999:7:::
```

```
sync:*:14214:0:99999:7:::
klog:*:14214:0:99999:7:::
messagebus:*:14214:0:99999:7:::
earthfans:$feivvg/fe0dff334sn/hbfwg.:14214:0:99999:7:::
dictd:*:14214:0:99999:7:::
mldonkey:*:14216:0:99999:7:::
mysql:!:14221:0:99999:7:::
Debian-exim:!:14222:0:99999:7:::
fetchmail:*:14222:0:99999:7:::
```

这些列的字段分别代表如下的意思。

- 账户名：密码：最后一次修改时间：最小时时间间隔：最大时间间隔：警告时间：不活动时间：失效时间：标志
- 账户名：与/etc/passwd 相对应。
- 密码：账户的密码，是经过加密存放的。长度应该是 13 个字符。而且应该只包含数字、字母与“{”、“}”、“/”字符。如果为“!”，代表该账户没有密码。
- 最后一次修改时间：表示从 1970 年 1 月 1 日起，到最后一次修改密码的天数。
- 最小时间间隔：表示密码在使用了指定的时间间隔后才能修改，单位是天。如为 10，则密码在使用了 10 天后才能修改。
- 最大时间间隔：表示密码可以使用的最大天数。
- 警告时间：表示密码在失效前几天给予用户警告，如为 10，则在密码失效前 10 天开始给出警告信息。
- 不活动时间：表示用户不登录多少天，账户仍然有效。如为 30，则表示用户 30 天不登录，账户仍然有效。
- 失效时间：表示账户的生存期，超过此期限，则该账户失效，无法进行登录。

21.4.3 组记录文件

用户的分组信息记录在/etc/group 文件，该文件的格式与前面两个文件相似，均是用“:”隔开的一行行文本。一个典型的 group 文件类似于这样：

```
root:x:0:
adm:x:4:earthfans
kmem:x:15:
dialout:x:20:earthfans
cdrom:x:24:earthfans
floppy:x:25:
tape:x:26:
earthfans:x:1000:
Debian-exim:x:130:
guest:x:1001:
```

该文件每行由 5 列组成，各列的意义如下。

- 组名：组的标识符。
- 密码：也可以给组设置密码。不过，一般 Linux 系统的组均没有密码。所以一般密码的

位置是“x”。

- 组号（GID）：组号，用数字来表示的组名。
- 用户列表：属于该组的用户列表。如果有多个用户，则用“,”隔开。

21.5　用户、组相关命令

系统中有许多的命令与用户、组相关，这些命令可以完成显示用户信息、修改用户等工作。下面分别予以介绍。

1．更改登录 Shell 命令 chsh

chsh 可以更改使用者的登录 Shell 设定，用法如下：

```
chsh -s shellname
```

shellname 可以是任何可运行的程序。但如果该程序不在/etc/shells 列表中，会给出一个警告信息。如果不使用参数，则进入交互式修改过程，会提示用户继续输入新的 shell 的名字，一个实际的命令使用过程如下：

```
chsh                                    #不加参数，进入交互式 Shell 修改过程
口令：                                  #输入密码
正在更改 earthfans 的 shell
请输入新值，或直接敲回车键以使用默认值
     登录 Shell [/bin/bash]:            #当前 Shell 为 bash，可以设置为其他值
```

2．显示用户信息命令 finger

finger 会显示指定用户的资料信息，如登录 Shell、计划、邮件等消息。而且还可显示远程主机上的指定用户的资料（不过，由于安全原因，现在大多数主机已经关闭了 finger 显示功能）。其使用方式是：

```
finger user[@host]
```

如“finger westnorth”则是显示 westnorth 用户的资料。下面是 finger 命令的输出信息示例：

```
Login: westnorth                                #用户的登录名
Name: Laptop Ma                                 #用户的全名
Directory: /home/westnorth                      #用户的主目录
Shell: /bin/bash                                #用户的登录 Shell
On since Sat Dec 13 06:55 (CST) on tty7 from :0
   6 hours 43 minutes idle                      #登录到本机的时间
No mail.                                        #没有邮件信息
No Plan.                                        #没有计划信息
```

如果不带参数直接运行 finger，会显示当前登录的用户信息：

```
finger                                          #不带参数运行 finger
Login     Name       Tty      Idle  Login Time   Office     Office Phone
earthfans  Laptop Ma   tty7     6:46  Dec 13 06:55 (:0)
```

如果使用类似“finger tom@23.202.198.111”会显示 IP 地址是 23.202.198.111 的主机上用

户“tom”的资料信息。

3．显示登录信息命令 last

last 可以用来显示本机的登录记录，如何时登录、何时登出等。单独使用时，命令读取/var/log 目录下的 wtmp 文件，显示其中关于用户登录的相关信息。命令的使用格式是：

```
last 参数 用户名
```

可用参数有：

```
-d：将 IP 地址转换为域名显示。
-R：不显示登录用户的主机名称。
-x：显示关机，重启等信息。
```

用法示例，如显示关机重启等信息使用“last -x”命令，如下所示：

```
last -x
earthfan pts/1         :0.0           Sat Dec 13 23:00   still logged in
earthfan tty7          :0             Sat Dec 13 06:55   still logged in
runlevel (to lvl 2)   2.6.27-10-generi Sat Dec 13 06:54 - 23:22  (16:27)
reboot   system boot  2.6.27-10-generi Sat Dec 13 06:54 - 23:22  (16:27)
reboot   system boot  2.6.27-7-generic Mon Dec  1 17:54 - 17:56  (00:02)
wtmp begins Mon Dec  1 17:54:13 2008
```

4．显示当前登录用户命令 who

who 命令可以显示当前登录的用户信息。用法：

```
who [选项]... [文件|参数 1 参数 2 ]
```

选项可取如下的值。

- -a：显示所有信息，是-b、-d、--login、-p、-r、-t、-T、-u 选项的组合。
- -b：上次系统启动时间。
- -d：显示已死的进程。
- -H：输出头部的标题列。
- --login：显示系统登录进程。
- --lookup：通过 DNS 查询主机名。
- -p：显示由 init 进程产生的活动进程。
- -r：显示当前的运行级别。
- -t：显示上一次系统时钟的变动。
- -u：列出已登录的用户。

下面是几个使用实例：

```
earthfans@mylaptop:~$ who -a           #显示所有信息
         system boot  2008-12-13 06:54
         run-level 2  2008-12-13 06:54                last=
LOGIN     tty6        2008-12-13 06:54             4502 id=6
LOGIN     tty5        2008-12-13 06:54             4493 id=5
LOGIN     tty4        2008-12-13 06:54             4492 id=4
earthfans + tty7        2008-12-13 06:55 08:06       6078 (:0)
```

```
earthfans + pts/1       2008-12-13 23:00   .        23136 (:0.0)
earthfans@mylaptop:~$ who -H           #显示当前登录用户的信息，同时显示标题
NAME     LINE        TIME           COMMENT
earthfans tty7        2008-12-13 06:55 (:0)
earthfans pts/1       2008-12-13 23:00 (:0.0)
earthfans@mylaptop:~$ who              #显示当前登录用户的信息，但不显示标题
earthfans tty7        2008-12-13 06:55 (:0)
earthfans pts/1       2008-12-13 23:00 (:0.0)
```

5．更改文件、目录的用户及组权限

可以通过命令 chown 来更改文件或目录的所属用户或组。其用法是：

```
chown [选项]... [所属用户][:[组]] 文件或目录...
```

选项可以取下列的值。

- -h：如果处理的目录或文件是符号链接，则只更改符号链接本身，而不更改‰链接指向的文件。
- “--from=user:group”：如果文件或目录当前的所有者是 user、所属组是 goup 时，执行更改。否则不更改文件目录的权限。这个选项比较适合于更改特定用户的文件，如 sam 离职，将其工作移交给 tom，则可使用--from=tom 来指定只修改原来属于 tom 的文件。
- -R：递归处理目录及文件。

下面是几个该命令的示例操作。

将/home/old 的属主更改为 new。可用如下命令：

```
chown new /home/old
```

将/home/old 的属主更改为 new，与此同时，也将该目录的所属组更改为 develop。使用如下命令：

```
chown new:develop /home/old
```

将/home/old 目录及其中所有的文件目录属主全部更改为 new，如果遇到符号链接文件，则只更改符号链接的权限。使用如下命令：

```
chown -hR new /home/old
```

21.6　小结

本章介绍了进行用户与组管理的一些方法，以及用户、组相关的实用命令，同时为高级用户介绍了用户、组的配置文件。对于普通用户来说，建立用户账号与编辑用户账号的功能比较实用、重要。而对于系统管理员而言，批量添加用户的功能由于可以进行批量操作，是比较实用的。对于系统管理员而言，读懂各配置文件所代表的意义也是很重要的。

第 22 章　系 统 安 全

随着电脑与生活的相关程度越来越大，很多重要资料与信息都会在电脑中存储与处理，所以保持电脑系统的安全变得越来越重要。虽然相对来说 Linux 的安全性能是比较好的，但还是应该了解一些 Linux 系统的安全知识，以更好地保护敏感数据。本章将介绍一些 Linux 的安全知识与防护技巧。

22.1　Linux 的良好安全系统

在介绍 Linux 的基础知识时，我们提到 Linux 的结构决定了 Linux 的安全性能是比较强劲的，但是，凭借其强大的安全性能可不可以做到完全对病毒与攻击免疫？本节讨论了这一主题。

Linux 的多用户管理方面，对于权限的分配无疑是做得极好的。在得出 Linux 是否安全这个结论前，先来回顾一下 Linux 系统关于文件的几个特性。

- 每一个文件分为三级权限，分别为文件的属主权限，属组权限及其他用户权限。也就是说，同一个文件，对于不同的用户有非常不同的权限。
- 应用程序一般并不放置在用户的主目录，而是被放置在“/bin”、“/sbin”或“/usr/bin”、“/usr/sbin”等几个目录。普通用户对于这些目录没有写的权限，只有读与执行的权限。
- 对于普通用户而言，要执行修改系统设置的任务、或者要由用户态进入内核态，必须切换到 root 权限。而切换到 root 权限的前提是要输入权限提升密码。
- 每个用户有自己的主目录，在自己的主目录内，拥有全部的权限，但对于其他的用户目录而言，一般只有读与运行的权限，没有写的权限。

从系统安全方面来说，这些均是极好的特性。如果用户主目录内一个可执行文件被感染了病毒，由于 Linux 具有如上特性，该病毒的影响范围也会被限制在该用户的主目录内。因为要修改系统文件，必须经由用户输入权限提升密码，这无疑极大地提高了病毒传染的难度。对于病毒而言，失去了传染性，也就失去了绝大部分的威胁。

而且，对系统了解越少，获得的权限也就越小，这样对于初学者来说，也就减少了许多病毒传染的可能。而对于经验丰富的系统管理员而言，虽然其有很高的系统权限，但管理员会密切注视系统文件，发现有异常会及时检查来保证安全。这也是一般的安全手册上一再要求 Linux 用户尽量不要使用 root 账号进行登录的原因所在。

病毒还有一个重要传播途径是通过软件下载。这一点在 Linux 系统中又得到了坚强的阻击。Linux 是建立在开源基础上的一个自由操作系统。同样，其上的绝大部分软件也同样是开源自由的。由于任何用户均可查看源代码。所以通过程序隐藏病毒的企图会从源头上得到制止。编译好的软件一般均提供相应的验证码，如 MD5 码来验证应用程序的完整性，所以将病毒编译进其他程序进行传播的企图又会被消灭在萌芽状态。

由于 Linux 系统文件所具备的特性，以及 Linux 文化中具备的阻止病毒传播的血统，使得病毒在 Linux 系统中生存与传播极为困难。曾经出现过的 Linux 病毒大多也是实验性质的，均为昙花一现，无法形成大规模的散播。

22.2　FireStarter 防火墙软件

大致了解了 Linux 的安全机制后，就应该考虑如何进行安全防护了。一般来说，对 Linux 威胁较大的属于网络入侵，所以只要对计算机的端口进行保护，就可以很大程度上提高 Linux 系统的安全性了。本章将介绍一款强大的防火墙软件 FireStarter。其是一款图形界面防火墙软件，功能强大、使用简单、提供了多种设置向导，可以方便地进行各种设置，是中小型网络与个人用户首选的防火墙软件。而且由于 FireStarter 防火墙实际是对 iptables 进行设置的，所以其设置的规则等极为透明，可以通过相关命令进行查看与修改。

22.2.1　安装、使用向导进行设置

Ubuntu 默认并未安装 FireStaret，要使用该软件，应该先进行安装。由于在 Ubuntu 的软件仓库中有该软件，所以可以简单地通过“sudo apt-get install firestarter”或在“新立得”中进行安装。具体安装方法，可以参见本书前面介绍的内容，这里就不再详细说明了。

但安装后由于未进行任何设置，所以其并不能满足普通用户的需求。在使用前，可以参照如下步骤进行设置。

（1）通过单击主菜单“应用程序”|“网络软件”|“FireStarter”来启动，也可以在终端中输入“firestarter”来启动。

（2）首次启动后，FireStarter 会弹出设置向导，用来设置一些系统基本信息。欢迎页面如图 22.1 所示。



图 22.1　FireStarter 设置向导

单击“Forward”按钮，到下一步“Network device setup”（网络设备安装）对话框。

（3）“Network device setup”（网络设备安装）对话框如图 22.2 所示，显示了当前检测出的网络设备。



图 22.2　设置向导之选择网络设备

（4）“Detected device（s）”（检测到的设备）下拉框显示了所有可连接到网络的设备，在其中选择要连接到网络的设备即可。如 eth0 代表第一个网卡。

（5）勾选“Start the Firewall on dial-out”复选框的意思是拨号后就开启防火墙。

（6）勾选“IP address is assigned via DHCP”复选框，意思是 IP 地址由 DHCP 动态指定。按照需要选择，然后单击“Forward”按钮到下一步“Internet connection sharing setup”（网络连接共享设置）对话框。

（7）“Internet connection sharing setup”（网络连接共享设置）对话框如图 22.3 所示。

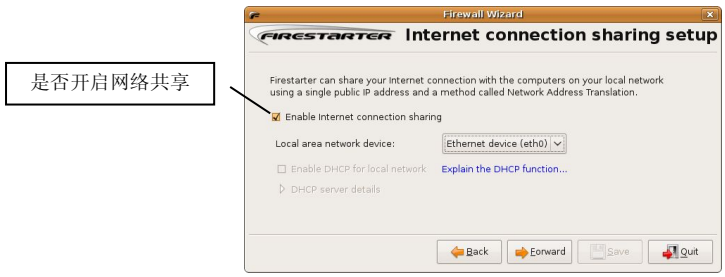


图 22.3　设置向导之共享上网

如果勾选了“Enable internet connection sharing”复选框，则开启了网络共享功能。这样的话，若此计算机连接到互联网，则连接到此计算机上的其他计算机也可以共享上网。按照自己的需要设置后，单击“Forward”按钮到下一步。

（8）现在向导就完成了。如果需要立即启动 FireStarter，则勾选“Start Firewall now”复选框，然后单击“save”按钮，完成设置，会按照向导设置启动防火墙服务。

22.2.2　基本设置

FireStarter 的界面如图 22.4 所示。可以从主界面直观地查看当前连接情况，并且可以方便地设置网络过滤器。界面各部分的大致功能也标注在图上。

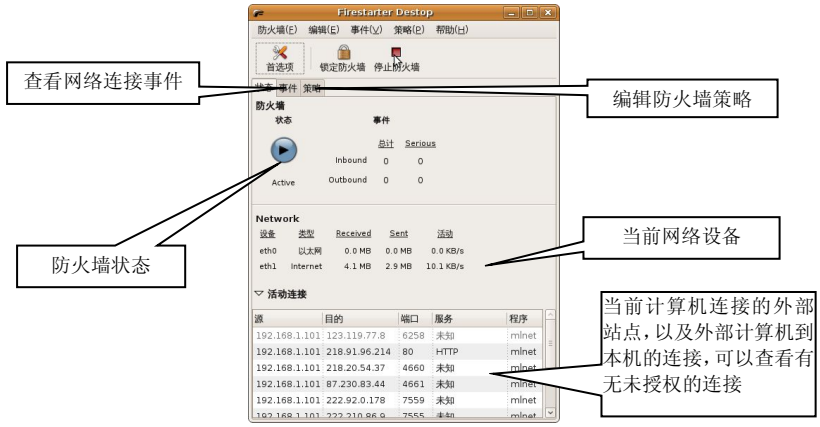


图 22.4　FireStarter 主界面

除了使用向导外，还可单击工具栏“首选项”按钮进行相关设定。在首选项中，除了可以设置向导中设置的几项内容外，还可对 ICMP、TOS 过滤进行设置。“首选项”的设置界面如图 22.5 所示。

22.2.3　策略设置

完成了软件的基本设置后，就可以开始进行防火墙最为重要的策略设置。单击图 22.4 所示“编辑防火墙策略”标签页或单击菜单“策略”|“添加策略项”命令，就可打开策略设置功能。

策略分为两种，一种是接收策略，管理其他计算机连接到本机的策略，如允许某些计算机连接到本机、不允许某些端口的请求访问本机等。另一种是发送策略，是指本机连接到其他计算机的策略，如允许连接到某 IP 地址、拒绝使用本机某端口进行连接等。两种策略的管理界面如图 22.6、图 22.9 所示。两种策略的切换通过“编辑”下拉框来实现。下面以几个例子说明其具体用法。



图 22.5　“首选项”设置界面

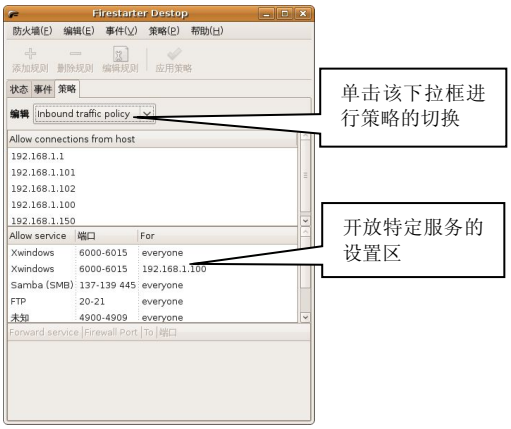


图 22.6　接收策略管理

1．允许某 IP 的计算机连接到本机

有时为了某些需要，可能需要允许某外部计算机连入本机。这时就可建立适当的策略来控制连入本机的计算机的 IP 地址。具体的设置步骤如下所示。

（1）由于是限制进入本计算机的计算机，属于接收策略，所以应该在如图 22.6 所示对话框“编辑”下拉框中选择“Inbound traffic policy”，进入接收策略编辑页面。

（2）出现如图 22.6 所示页面，在“Allow connection from host”列表框中单击鼠标右键，在弹出的菜单中选择“添加规则”命令。

（3）弹出“Add new inbound rule”（添加接收规则）对话框，如图 22.7 所示。



图 22.7　添加接收规则

在“Allow connections from IP,host or network”文本框中，输入允许连接的 IP 地址，如“192.168.1.100”，在“Comment”文本框中输入该 IP 地址的注释。设置结束后，单击“Add”按钮添加规则。

（4）添加所有允许的 IP 地址后，单击如图 22.6 所示“应用策略”按钮应用所做的设置后，“192.168.1.100”就会被允许连接到本机。而其他未授权的 IP 地址所发送的连接请求就会被拒绝。可以单击图 22.6 所示的“事件”选项页查看详细信息。

2．开放本机的某项服务

在实际使用中，经常会遇到这种情况：不允许其他计算机连接到本机，但允许访问本机所提供的某项服务，如 HTTP、SSH 等。这时，就开放这些服务所需要的端口，这些均包含在接收策略中，而且开放某些端口也是很容易的。详细步骤如下所示。

（1）参照上述步骤，进入“接收策略”编辑页。



图 22.8　允许某项服务所用的端口

（2）在如图 22.6 所示接收策略管理窗口中，在“Allow Service”区域单击鼠标右键，选择“添加规则”菜单项，出现如图 22.8 所示对话框。

（3）在“名称”下拉框中，选择要开放的服务，如“Samba”，端口中会自动出现该服务所使用的端口，如果用户已经对该服务的默认端口进行了修改，也可以自行输入实际的端口数值。

（4）在“When the source is”区中设置开放的对象，可以为“任何人”，也可以是特定的 IP 地址。

（5）设置结束后，单击“Add”按钮就开放了该服务。其他计算机就可以使用本机提供的该服务了。

3．拒绝本地计算机连接到某些 IP 地址

如果将安装了 Firestarter 的计算机作为网关或者防火墙使用，则有可能需要限制本地网络的计算机，使它不能连接到某外部 IP 地址。这些工作，可在如图 22.9 所示的发送策略中进行处理。

发送策略有两种工作方式，一是白名单，另一种是黑名单。下面是其概念的介绍。

- 白名单：是指默认不允许连接到所有的 IP 地址，然后列出一个白名单，只允许连接到白名单中的 IP 地址。这种方式安全性高，但是对于连接的限制也比较大。
- 黑名单：是默认允许连接到所有 IP 地址，然后列出一个黑名单，不允许连接到黑名单列出的 IP 地址。这两种方式各有其使用场合，不过，对于普通用户而言，最常用的是黑名单方式。

下面介绍使用黑名单方式进行设置的步骤，白名单可参照其进行设置：

（1）图 22.9 所示发送策略有两个选项。一是“permissive by default,blacklist traffic”，这是黑名单方式。还有一个是“Restrictive by default,whitelist traffic”，这是白名单方式。选择黑名单，说明下文所述添加的 IP 地址就是不允许访问的 IP 地址，而选择白名单则刚好相反，添加的 IP 地址是允许访问的 IP 地址。本例中选择“黑名单”方式。

（2）在“Deny connection to host”（拒绝连接的 IP 地址）列表框中单击鼠标右键，在弹出

的菜单中选择“添加规则”菜单项，出现如图 22.10 所示的对话框。

（3）在“Deny connections to”文本框中输入不允许访问的网站。单击“add”按钮，就添加了一个不允许访问的网站。

（4）单击如图 22.9 所示界面中“应用策略”按钮，便完成了本次设置。

（5）打开浏览器，输入刚才拒绝连接的网站地址（如 www.badguy.com），就会发现已经无法连接了。

参照此方法，还可以将某项服务也设置为不允许某些用户访问。可以参见图 22.10，而且设置方法与前述类似，在此就不再赘述了。

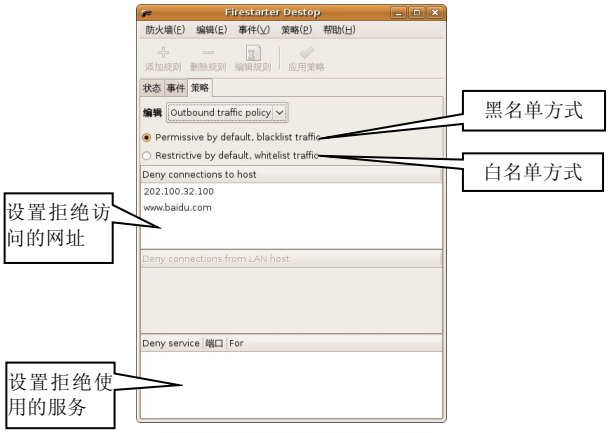


图 22.9　发送策略管理



图 22.10　不允许连接到某网址

4．查看 FireStarter 设置的 iptables 表

前面介绍过，FireStarter 的相关设置实际上是对 iptables 进行设置。所在在设置了相关的策略后，就可以通过 iptables 的命令查看了。iptables 显示策略的参数是-L。命令如下所示：

```
    $sudo iptables -L                    #显示当前计算机中采用的策略
    ..........
    #下面是接收策略，可与 FireStarter 中的设置对应查看
    Chain INBOUND (1 references)
    target     prot opt source               destination
    ACCEPT     tcp  --  anywhere             anywhere            state RELATED,
ESTABLISHED
    ACCEPT     udp  --  anywhere             anywhere            state RELATED,
ESTABLISHED
    ACCEPT     all  --  192.168.1.1          anywhere
    ACCEPT     all  --  Destop.www.earthfans.cn  anywhere
    ACCEPT     all  --  192.168.1.102        anywhere
    ACCEPT     all  --  192.168.1.100        anywhere
    ACCEPT     all  --  192.168.1.150        anywhere
    ACCEPT     tcp  --  anywhere             anywhere            tcp dpts:x11:6015
    ACCEPT     udp  --  anywhere             anywhere            udp dpts:x11:6015
    ACCEPT     tcp  --  192.168.1.100        anywhere            tcp dpts:x11:6015
    ACCEPT     udp  --  192.168.1.100        anywhere            udp dpts:x11:6015
```

```
    ACCEPT     tcp  --  anywhere            anywhere          tcp dpts:netbios-ns
    ……
    #下面是发送策略
    Chain OUTBOUND (1 references)
    target     prot opt source               destination
    ACCEPT     icmp --  anywhere             anywhere
    ACCEPT     tcp  --  anywhere              anywhere            state RELATED,
ESTABLISHED
    ACCEPT     udp  --  anywhere              anywhere            state RELATED,
ESTABLISHED
    LSO        all  --  anywhere             202.100.32.100
    LSO        all  --  anywhere             220.181.6.19
    LSO        all  --  anywhere             220.181.6.18
    ACCEPT     all  --  anywhere              anywhere
```



通过 FireStarter 所设置的接收与发送策略，包括允许、禁止的服务及 IP 地址均可通过 iptables 的相关设置进行实现。但相对来说比较复杂，不推荐初学者使用该方法。

22.3 PAM 机制

PAM 机制是 Linux 系统安全机制的根本，所有需要验证的场合基本均使用了 PAM 验证方式。为了对系统安全机制有比较深入的理解，有必要了解这个重要的安全机制。本节就来介绍一些 PAM 机制的简单知识。

22.3.1 PAM 原理

系统安全的一个重要方面就是验证用户身份。登录系统时，会要求输入用户名与密码，输入正确才会允许用户登录。这样可以阻止非授权用户进入系统进行破坏性的操作。其他系统也是一样，如电子邮件系统也是通过邮箱地址与密码匹配的方式来进行认证的，只有通过验证的用户才被认为是用户本人，然后会允许其查看邮件和发送邮件等。

这些过程看起来似乎很简单。只需要做一个表格，一列放置用户名，一列放置密码。在接到验证身份请求时，直接查表，如果用户名与密码均与表中内容相符，则通过验证，否则不能通过验证。但是，这种方式一方面存在着很大的安全隐患。首先，存放着明文的密码表格一旦被黑客获得，就会轻易获得登录授权，其次系统管理员后有可能会利用用户名与密码登录，查看用户的隐私材料。另一方面，这种方式不能提供更为强大的登录验证手段，如利用生物识别信息的验证，也无法提供更为复杂的登录管理手段，如限制密码错误输入次数等。

然后考虑到这些因素，许多系统的验证采用了一系列复杂的算法来保证密码的安全性，增加了许多功能用来完成复杂的管理工作。这样虽然达到了比较满意的效果，但是对于操作系统层面的验证来说，有一个致命的缺陷，就是这种验证方式缺乏灵活性与扩展性。如果一个验证密码的方式出现了安全漏洞或者需要更改验证方式，就不得不重新编译整个程序。而且一个软件的密码验证方式就算再安全再成功，其他软件系统也不能或很难分享其功能。

1995 年 Sun 公司提出了 PAM 的概念，从很大程度上解决了操作系统层面验证用户信息的种种问题。PAM 的英文全称为 Pluggable Authentication Modules，笔者将其译成“即插即用型验证模块”。其基本思想是编写一些用来完成验证工作的库文件，然后通过一套公开的 API 来调用这些库完成验证。至于验证中要使用哪些库，则交由 PAM 设置文件来定义。在 Ubuntu 系统中，完成验证功能的库文件存放在/lib/security 目录，而 PAM 的配置文件是/etc/pam.conf 或者/etc/pam.d/中的文件。

22.3.2　PAM 配置文件

Linux 系统中 PAM 的原理如前所述，本小节介绍一些 PAM 中重要的配置文件。因为这些配置文件决定了各项服务使用何种 PAM 方法。

前面提到过，PAM 的配置文件主要是/etc/pam.conf 或者/etc/pam.d/中的文件。最初的做法是将所有的配置信息写入/etc/pam.conf。但这样做的后果是该文件会越来越大，越来越复杂，不利于对其进行管理。比较新型的 Linux 发行版通常会将 PAM 配置文件写入/etc/pam.d/中一个个独立的文件中，每项服务所用的 PAM 配置写到以该服务命名的独立的文件中。使用 ls 命令查看这些 PAM 文件，其概况是：

```
earthfans@mylaptop:~$ ls /etc/pam.d/   #使用 ls 命令查看 PAM 目录中的配置文件
atd            common-auth      cups              login   ppp    wu-ftpd
chfn           common-password  gdm               other   samba  xscreensaver
chsh           common-session   gdm-autologin     passwd  su
common-account  cron            gnome-screensaver  polkit  sudo
```

该目录中的 PAM 配置文件，以 common 开头的是通用配置文件，以备其他配置文件调用；other 是默认配置；而其他配置文件则是相关服务的配置，如 wu-ftpd 是 ftp 服务器的 PAM 配置等。下面以“login”文件为例介绍配置文件的格式。

```
$cat /etc/pam.d/login                          #显示 login 文件内容
#底层模块类型  控制标志                        加载模块名          参数
auth       requisite  pam_securetty.so
auth       requisite  pam_nologin.so
session    required   pam_seLinux.so close
session       required   pam_env.so readenv=1
session       required   pam_env.so readenv=1 envfile=/etc/default/locale
@include common-auth
auth       optional   pam_group.so
session    required   pam_limits.so
session    optional   pam_lastlog.so
session    optional   pam_motd.so
session    optional   pam_mail.so standard
@include common-account
@include common-session
@include common-password
session required pam_seLinux.so open
```

配置文件内容详解如下。

第一列是“底层模块类型”，代表 PAM 执行的任务类型。因为每一个 PAM 模块可能包含多种类型的验证选项，指定 PAM 类型，则只执行相应模块相应功能。底层模块有四种类型，分别说明如下。

- auth：即“authentication”。该类型的模块执行的任务是识别用户身份。一般情况，执行的动作是要求输入用户名与密码，但也可用智能卡等设备进行验证。
- account：对账号的属性进行检查，如密码是否到期，该用户是否有使用某项服务的许可等。
- session：指定用户登录前与退出后要进行的操作。如主目录的挂载、用户信息的核查等。
- password：主要完成用户密码的更新。

第二列是“控制标志”，主要是根据 PAM 模块执行的返回信息进行判断，然后做相应的动作。它也有四个选项。

- required：需要所有此标记的模块返回值为成功。但即使接到模块返回的失败值也不立即退出，而是继续执行同类型的下一个模块，所有模块均执行完后，才向应用程序返回失败值。
- requisite：与 required 基本类似，但有一点区别，就是只要收到模块返回的失败值则立即宣告模块执行失败，不执行其他模块，立即向应用程序返回失败值。
- sufficient：接到该选项中的任一模块返回成功值后，立即向应用程序返回成功值，不再进行后面模块的检查。如果返回错误，则该选项功能与 optional 相同。
- optional：该选项说明 PAM 模块不进行返回正确与否的检查，一般只是返回一个 PAM_IGNOR 信号。

第三列是“加载模块名”。指定了要调用的模块的路径与名称。一般并不显式指定路径，因为系统默认模块被放置在/lib/security/目录中，只有放置在其他目录的模块才需要显式指定目录。所以，一般情况下，第三列只指定了模块名。

第四列是“参数”，是传递给模块的参数，可以有多个参数，之间用空格分隔开即可。

综合上面的介绍，就可以理解这个 PAM 配置文件了。第 1、2 行说明“login”服务（也就是系统登录服务），使用了 pam_securetty.so 与 pam_nologin.so 验证用户身份，这是由关键字“auth”指定的。而且这两个难信息如果出错，则立即退出，不再执行其余的 PAM 操作，这个行为由关键字“requisite”指定。

第 7 行还有一个验证操作，不过该验证操作不太重要，因为不管其结果正确与否，均不会对其他操作有什么影响。配置文件中还有大量的 seesion 关键字，这是在用户登录前完成某些设置工作，如“session required pam_env.so readenv=1”是用来读取环境变量，而且该操作必须得到正确的返回值。

同时，在配置文件中还有一些前面没有介绍的关键字“@include”这是用来包含一些通用的PAM配置文件的命令，如“@include common-account”就相当于将配置文件“common-account”文件中的内容插入到该处。这种方式对于可能多次重复使用的代码比较有用。

了解了配置文件的内容规则后，就可以为自己的服务编写 PAM 配置文件了。方法也很简单，在“/etc/pam.d”目录中新建一个与自己的服务同名的文件，如服务名为“myrule”，则建立一个名为“myrule”的文件，然后在文件中按照前述内容填写相应的 PAM 规则即可。

22.4 使用 seahorse 进行文件加密

数据加密是计算机安全体系中非常重要的一环，广义的数据加密同时也包含了用户的身份认证与数字签名等内容。本节简单地介绍了 Ubuntu 系统中的一款强大的 PGP 加密软件 seahorse

的使用。

PGP 全称很有趣，是“Pretty Good Privacy”。是一个基于 RSA 非对称加密算法的密码系统。其结合了 RSA 公钥体系的安全便捷和对称算法的速度，在数字签名和密钥认证管理机制上的设计很精巧，因此 PGP 在非对称加密体系中逐渐星露头角。Ubuntu 系统自带了一个提供 PGP 密码服务的软件 seahorse。更重要的是，使用其进行加、解密文件与文件夹也是很简单的。

22.4.1　创建密钥

在使用 seahorse 进行 PGP 方式的加密前，需要先创建一个用于加密的密钥。创建密钥可分为如下五个步骤。

（1）单击主菜单“应用程序”|“附件”|“密码与密钥管理”命令或者在终端中输入“seahorse”打开密钥管理器，其界面如图 22.11 所示。

（2）单击 seahorse 菜单“密钥”|“创建密钥”命令，打开了“创建新密钥”对话框，如图 22.12 所示。

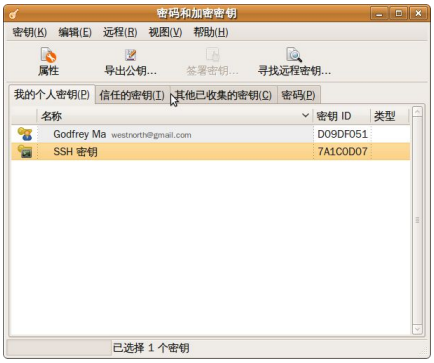


图 22.11　seahorse 主界面

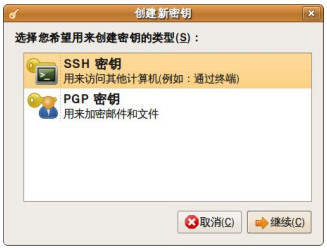


图 22.12　选择密钥种类

（3）由图可知，此时可创建两种密钥，一种是用于 SSH 连接的密钥，还有一种就是 PGP 密钥。选择 PGP 密钥，单击“继续”按钮，出现如图 22.13 所示的密钥设置对话框。

（4）在其中填写适当的内容，其中加密类型可以选择 DSA EIGamal，然后单击“创建”按钮。

（5）现在会提示输入密码，该密码在使用 PGP 加密与解密时均要使用，需要牢记。输入密码后，就开始创建密钥的过程。这可能需要等待稍微长一些的时间，对于普通个人计算机而言，可能需要 5～10 分钟。创建密钥完成后，就可使用该密钥对文件进行加密了。



图 22.13　创建新 PGP 密钥

22.4.2　文件加密与数字签名

创建了密钥后，就可以使用这 PGP 密钥对文件进行加密了。加密的方法很简单，按如下步骤操作即可。

（1）在文件浏览器中定位到要加密的文件，然后右击该文件，选择“加密”菜单项，会弹出图 22.14 所示的对话框。

（2）选择要使用的 PGP 密钥，还可选择是否给加密文件添加数字签名，设置结束后，单击“OK”按钮，开始对文件进行加密。

（3）系统会提示要求输入 PGP 的密码，输入创建 PGP 密钥时设置的密码，就可以完成文件的加密了。

加密后的文件名为原文件加上 pgp 的后缀，如加密“test.txt”文件，则加密后的文件名为“test.txt.pgp”。

加密目录的方法基本与加密文件的类似，不过，会在上述步骤（1）前弹出图 22.15 所示的对话框，要求用户进行选择。如果选择“Encrypt each file separately”则会在该目录中为每一个文件生成一个加密后的文件。而如果选择“Encrypt packed together in a package”，则会先将该目录进行压缩，然后加密该压缩文件。



图 22.14　加密文件



图 22.15　加密目录时弹出的提示

22.4.3　添加数字签名

在文件管理器中，右击任一文件，选择“Sign”菜单项，就可以给文件添加数字签名。在添加数字签名的过程中，需要输入 PGP 密钥的密码。

添加数字签名有两个目的，一是确认用户身份，另一个是确保文件不被非法篡改。给文件添加的数字签名是一个小文件，文件名是原文件名加上“.sig”的后缀。

要验证数字签名，只需将原文件与数字签名文件放置于同一个目录中，然后双击数字签名文件，即可完成验证。验证结果会弹出一个窗口，显示在屏幕的右下侧。如果验证失败显示的浮动框如图 22.16 所示。有两种可能，一是该文件并非来自声明的创建者，二是该文件已经经过了篡改，不再是原始内容了。如果验证成功，则显示图 22.17 所示的对话框，说明文件来自声明的创建者，而且没有经过篡改。



图 22.16　验证失败



图 22.17　验证成功

22.4.4　公钥管理

由于 PGP 是一种非对称的加密技术，所以其创建的密钥是一个密钥对：用来加密的公钥与用来解密的私钥。为了更好使用需要对其进行有效管理。

1．分发公钥

为了可以让其他用户也用该密钥加密文件，就需要将此 PGP 密钥中的公钥分发给其他用户。在分发前，需要先将公钥内容导出。导出公钥的步骤如下所示：

（1）在 seahorse 的主界面选择工具栏中的“导出公钥”按钮；

（2）在弹出的对话框中定位到保存公钥的目录，单击“确定”将公钥导出。

一般情况下，公钥名称是 PGP 密钥的名称，后缀为 asc。导出后，就可将其分发给其他要发送文件给你的用户，使这些用户能使用该公钥加密文件。

2．导入、签署公钥

收到其他用户分发的公钥后，还需要导入并签署该公钥才能使用该进行加密工作。导入公钥的方法是单击主界面“编辑”菜单中“导入”菜单项，然后定位到接收的公钥文件，就可完成导入工作。但一个公钥导入后，并不能立即投入使用，还需要进行签署才可正常使用。签署公钥的步骤如下所示。

（1）在如图 22.11 所示的主界面中单击“其他已收集的密钥”标签，进入该标签页。

（2）单击选中标签页中列出的公钥，单击工具栏的“签署”按钮，出现如图 22.18 所示的“签署公钥”对话框。

（3）选择相应选项后，单击“签署”即可完成签署。



图 22.18　签署公钥

公钥被签署后，就可使用该公钥进行加密操作了。有一点要注意的是，使用公钥进行加密后，加密文件的用户是无法对其进行解密的。只有掌握与该公钥配对的私钥的用户才能对其解密。还有一点，在签署公钥前，系统中必须要存在一个 PGP 密钥，如果系统中没有的话，应该先行创建，然后才可进行密钥签署工作。



在传输重要及敏感数据的时候，可以采取 PGP 加密，这是一种高安全级别的加密算法，具有极高的安全性。如果再加上数字签名，就更能保证数据的安全性，同时防止非法篡改。也可以将公钥存于公用网络，以方便随时进行数据的加密与数字签名。

22.5　了解与分析日志文件

Linux 中一项重要的监视系统性能与系统安全的工具是 syslog，其可将各种不同的程序所产生的日志输入到特定的设备，以供查看。syslog 的核心是 syslogd 守护进程，这是用来接收并处理日志信息的程序。还有一个重要的配置文件是/etc/syslog.conf，用来设置 syslogd 处理各种日志的方式。

22.5.1　解读/etc/syslog.conf 配置文件

系统所产生的日志可分为多个类型，而每个类型的日志又可分为多个优先级别。syslogd 还可对这些日志进行不同的处理，如写入文件或者直接送往打印机等。这些操作均由 syslog.conf 定义。一个典型的/etc/syslog.conf 文件内容应该如下所示，这里只列出其实际设置内容，实际

的文件中还有许多以“#”开头的注释部分：

```
$cat /etc/syslog.conf
auth,authpriv.*         /var/log/auth.log
 *.*;auth,authpriv.none     -/var/log/syslog
#cron.*             /var/log/cron.log
daemon.*            -/var/log/daemon.log
kern.*              -/var/log/kern.log
lpr.*               -/var/log/lpr.log
mail.*              -/var/log/mail.log
user.*              -/var/log/user.log
mail.info           -/var/log/mail.info
mail.warning            -/var/log/mail.warn
mail.err            /var/log/mail.err
news.crit           /var/log/news/news.crit
news.err            /var/log/news/news.err
news.notice         -/var/log/news/news.notice
 *.=debug;\
auth,authpriv.none;\
news.none;mail.none-/var/log/debug
 *.=info;*.=notice;*.=warning;\
auth,authpriv.none;\
cron,daemon.none;\
mail,news.none      -/var/log/messages
 *.emerg                *
daemon.*;mail.*;\
news.err;\
*.=debug;*.=info;\
*.=notice;*.=warning    |/dev/xconsole
```

可以看出，配置文件由多行配置组成，但各行均遵循一定的格式。该格式一般类似于：

```
facility.level action
```

- facility：用来指定要记录的日志类别。
- level：用来指定所记录日志的级别。
- “facility .level”被称为选择器。
- action 是选择器所执行的操作。

可以给多个 facility 指定一个级别，只要将多个 facility 之间用“,”隔开即可。同样也可以使多个选择器执行同一个操作，只要将各个选择器用“:”隔开即可。

facility 包括以下这些类别，意思是由如下程序或活动产生的日志。

- auth：系统登录信息，如 login，su，getty 等产生的日志。
- authpriv：包括特权用户在内的登录信息。
- cron：与 cron 和 at 等进程自动调度产生的相关日志。
- daemon：其他守护进程产生的日志。
- kern：内核信息。

- lpr：与打印服务有关的日志。
- mail：与电子邮件有关的日志。
- mark：生成时间戳。
- news：来自新闻服务器的日志信息。
- syslog：由 syslogd 生成的日志信息。
- user：由用户程序生成的日志信息。
- uucp：由 uucp 生成的日志信息。
- local0----local7：保留，可以对其进行自定义。如使用 local5 捕捉 ssh 的日志信息。
- *：通配符，代表除了 mark 以外的所有日志信息。

level 是指所记录日志的级别。这是由于每个 Facility 会产生多种级别的日志，所以需要指定需要记录的日志级别。指定级别则只有该级别或更高级别的日志会被记录。可用的级别如下。

- emerg：紧急情况。
- alert：需要立即被改正的错误信息。
- crit：重要情况。
- err：错误信息。
- warning，warn：警告信息。
- notice：不是重要情况，但需要引起注意。
- info：情报信息。
- debug：调试时产生的信息。
- none：没有重要级，通常用于调试。
- *：通配符，除了 none 的其他所有级别。

每个功能对应的优先级是按一定顺序排列的，emerg 是最高级，然后是 alert，其他的重要级别按上面列出的顺序依次递减。

action 字段是指得到符合前面设定的日志后所采取的动作。syslogd 可以支持如下动作。

- file：指定文件的绝对路径，将日志写入文件中，这也是日志文件最常用的动作。
- terminal：将日志送到终端显示。
- print：将日志送往打印机。
- @host：将日志写到远程的日志服务器。
- username：向同时登录到系统的另一个用户发送信息。
- namedpipe：使用命名管道文件。

了解了这些知识后，就可以理解 syslog.conf 文件了。如：

```
auth,authpriv.*          /var/log/auth.log
```

意思是将所有级别（由“*”指定）的系统验证信息（由“auth，authpriv”指定）写入到文件“/var/log/auth.log”中。其他的内容也可以参照进行分析理解。

22.5.2　syslogd 守护进程的参数

syslog 守护进程是由/etc/rc.d/init.d/syslog 脚本启动的，然后通过 2 级运行级别的脚本“/etc/rc2.d/S10sysklogd”来调用。查看该脚本就可以发现，syslog 的运行默认是没有指定任何参数的。但其实 syslogd 还是有一些比较有用的参数的。例如：

- -r：监听 514 端口，该端口是用来发送日志信息的。使用这个参数的计算机可以作为远

程日志服务器，接收其他计算机的日志信息；

- -h：转发远程计算机的日志信息。默认是不进行转发的。

要添加 syslog 的参数，应该打开“/etc/default/syslogd”，其内容类似这样：

```
SYSLOGD=""
```

然后将相应的参数添加进去。加入后其内容应该如下所示：

```
SYSLOGD="-rh"
```

22.5.3 查看日志文件

理解日志文件的目标是为了查看系统日志，以便分析系统的错误或者是检查系统的安全程度。所以，在了解了如何产生日志文件后，应该再来查看与分析日志文件。

通过分析 syslog.conf，可以知道系统验证日志信息写在/var/log/auth.log 中。而其他系统信息写入了 syslog 文件，用户的日志信息写入了 user.log 文件。对于普通用户而言，比较重要的日志文件也就这么几个。下面分别简单地查看其内容。

1. syslog 日志

除了系统验证日志外的其他系统日志均记录在该日志中，其内容如下所示：

```
    $cat /var/log/syslog                    #使用 cat 命令查看该日志
    Dec 21 08:59:34 mylaptop syslogd 1.5.0#2ubuntu6: restart.
    Dec 21 08:59:34 mylaptop anacron[23865]: Job `cron.daily' terminated
    Dec 21 08:59:34 mylaptop anacron[23865]: Normal exit (1 job run)
    Dec 21 09:00:03 mylaptop kernel: [ 7113.200115] usb 5-1: new high speed USB
device using ehci_hcd and address 2
    Dec 21 09:00:03 mylaptop kernel: [ 7113.334072] usb 5-1: configuration #1
chosen from 1 choice
    Dec 21 09:00:03 mylaptop kernel: [ 7113.452073] usb 5-1: reset high speed
USB device using ehci_hcd and address 2
    Dec 21 09:00:04 mylaptop kernel: [ 7113.631022] ndiswrapper: driver zd1211bu
(Wireless,06/25/2007,6.22.0.0) loaded
    Dec 21 09:00:04 mylaptop kernel: [ 7114.347634] wlan0: ethernet device
00:21:67:e8:16:05 using NDIS driver: zd1211bu, version: 0x6160000, NDIS version:
0x501, vendor: '802.11 b+g Wireless LAN', 0ACE:1215.F.conf
```

可以从日志中看出，系统在 12 月 21 日，8 时 59 分 34 秒开始重启。然后自动调度进程也被中止。然后系统在 9 时 0 分 03 秒启动。在第一个 USB 接口上发现了一个高速 usb 设备。04 秒时使用 ndiswrapper 为该设备加载驱动程序。同时检测出该设备是芯片为 zd1211 的无线网卡。接着在系统中为其分配标志为 wlan0，该网卡设备号为 0acd:1215。

2. auth.log 日志

该日志中记录了系统验证信息。其内容如下所示：

```
    $cat /var/log/auth.log                  #使用 cat 命令查看其内容
    Dec 19 10:38:29 mylaptop gdm[5711]: pam_unix(gdm-autologin:session):
session opened for user earthfans by (uid=0)
    Dec 19 10:38:29 mylaptop gdm[5711]: pam_ck_connector(gdm-autologin:
```

```
session): nox11 mode, ignoring PAM_TTY :0
    ……
    Dec 20 09:57:15 mylaptop sudo: westnorth : TTY=pts/0 ; PWD=/home/westnorth ;
USER=root ; COMMAND=/bin/mv /etc/rc2.d/S20vboxdrv /etc/rc2.d/K20vboxdrv
    Dec 20 09:57:50 mylaptop sudo: westnorth : TTY=pts/0 ; PWD=/home/westnorth ;
USER=root ; COMMAND=/bin/mv /etc/rc2.d/S20virtualbox-ose /etc/rc2.d/K20
virtualbox-ose
    Dec 20 10:00:16 mylaptop sudo: westnorth : TTY=pts/0 ; PWD=/etc/rcS.d ;
USER=root ; COMMAND=/usr/bin/updatedb
    Dec 20 10:10:33 mylaptop sudo: westnorth : TTY=pts/1 ; PWD=/home/westnorth ;
USER=root ; COMMAND=/sbin/ifconfig wlan0 mtu 1450
```

从这几行日志中可以看到，12 月 19 日 10 时 38 分 29 秒开始通过 pam_unix，并进行自动登录（这是 Ubuntu 提供的一项服务，可以使用户不用输入用户名、密码即可完成登录，虽然有一定的安全隐患，但是却方便了许多）。12 月 20 日 9 时 57 分 15 秒，用户 westnorth 使用 sudo 命令提升权限到 root，然后执行了“mv /etc/rc2.d/S20vboxdrv /etc/rc2.d/K20vboxdrv”命令。后面也是几条使用 sudo 提升权限执行命令的记录。

3．user.log

这是用户在操作过程中的一些信息，主要是一些错误信息。其内容如下所示：

```
    $cat /var/log/user.log                    #使用 cat 查看其内容
    ……
    Dec 21 07:02:26 mylaptop pulseaudio[5862]: module-x11-xsmp.c: X11 session
manager not running.
    Dec 21 07:02:26 mylaptop pulseaudio[5862]: module.c: Failed to load  module
"module-x11-xsmp" (argument: ""): initialization failed.
    Dec 21 07:14:45 mylaptop logger: hello world
    Dec 21 11:18:59 mylaptop bonobo-activation-server (earthfans-9441): could
not associate with desktop session: Failed to connect to socket /tmp/dbus-
0ZcPIIalhO: Connection refused
    Dec 21 11:20:18 mylaptop pulseaudio[5817]: ltdl-bind-now.c: Failed to find
original dlopen loader.
```

user.log 是用户执行相应的操作时一些错误记录。其中有一行比较怪异，Dec 21 07:14:45 mylaptop logger: hello world。其实，这是使用 logger 命令向日志系统中手动写入的信息。这个命令的用法是：

```
    logger "I hack this machine,please contact zs@zhangsan.com"
```



通过 logger 命令可以看到，日志文件也不总是可靠的。而且攻击者如果攻入系统，总会在退出时改写相关的日志信息。所以，判断系统安全情况，还要综合其他方面内容来考量。

22.6 小结

本章介绍了一些对 Linux 系统进行安全设置的方法，包括防火墙软件的设置、PAM 机制、文件加密的方法、分析日志文件的方法。对于普通用户来说，FireStarter 防火墙软件的使用是比较重要也比较实用的内容，同时文件加密与添加数字签名的方法对普通用户也具有很强的实用价值，而且这些操作是比较简单的，读者可参照本章内容练习掌握。

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：（010）88254396；（010）88258888
传　　真：（010）88254397
E-mail：　dbqq@phei.com.cn
通信地址：北京市万寿路 173 信箱
　　电子工业出版社总编办公室
邮　　编：100036

《Linux 管理与开发实用指南——基于 Ubuntu》
读者交流区

尊敬的读者：

感谢您选择我们出版的图书，您的支持与信任是我们持续上升的动力。为了使您能通过本书更透彻地了解相关领域，更深入的学习相关技术，我们将特别为您提供一系列后续的服务，包括：

1．提供本书的修订和升级内容、相关配套资料；

2．本书作者的见面会信息或网络视频的沟通活动；

3．相关领域的培训优惠等。

请您抽出宝贵的时间将您的个人信息和需求反馈给我们，以便我们及时与您取得联系。

您可以任意选择以下三种方式与我们联系，我们都将记录和保存您的信息，并给您提供不定期的信息反馈。

1．短信

您只需编写如下短信：B09650+您的需求+您的建议

发送到1066 6666 789（本服务免费，短信资费按照相应电信运营商正常标准收取，无其他信息收费）

为保证我们对您的服务质量，如果您在发送短信24小时后，尚未收到我们的回复信息，请直接拨打电话（010）88254369。

2．电子邮件

您可以发邮件至jsj@phei.com.cn或editor@broadview.com.cn。

3．信件

您可以写信至如下地址：北京万寿路173信箱博文视点，邮编：100036。

如果您选择第2种或第3种方式，您还可以告诉我们更多有关您个人的情况，及您对本书的意见、评论等，内容可以包括：

（1）您的姓名、职业、您关注的领域、您的电话、E-mail地址或通信地址；

（2）您了解新书信息的途径、影响您购买图书的因素；

（3）您对本书的意见、您读过的同领域的图书、您还希望增加的图书、您希望参加的培训等。

如果您在后期想退出读者俱乐部，停止接收后续资讯，只需发送“B09650+退订”至10666666789即可，或者编写邮件“B09650+退订+手机号码+需退订的邮箱地址”发送至邮箱：market@broadview.com.cn亦可取消该项服务。

同时，我们非常欢迎您为本书撰写书评，将您的切身感受变成文字与广大书友共享。我们将挑选特别优秀的作品转载在我们的网站（www.broadview.com.cn）上，或推荐至CSDN.NET等专业网站上发表，被发表的书评的作者将获得价值50元的博文视点图书奖励。

我们期待您的消息！

博文视点愿与所有爱书的人一起，共同学习，共同进步！

通信地址：北京万寿路173信箱　博文视点（100036）　　电话：010-51260888

E-mail：jsj@phei.com.cn，editor@broadview.com.cn